DOCUMENT RESUME

ED 105 199		CE 003 487
AUTHOR	Weissmuller, Johnny J.; And Others	
TITLE	CODAP: Source Program Listings for the Univac 1108.	
INSTITUTION	Air Force Human Resources Lab., Lackland AFB, Tex. Computational Sciences Div.	
REPORT NO	AFHRL-TR-74-83	
PUB DATE	Oct 74	
NOTE	535p.; For related volumes see CE 003 485-486	
EDRS PRICE	MF-$ 0.92 HC-$27.29 PLUS POSTAGE	
DESCRIPTORS	Career Ladders; Comprehensive Programs; *Computer Programs; Informaticn Processing; *Information Retrieval; Information Services; Information Sources; *Information Systems; Job Analysis; Occupational Clusters; *Occupational Information; Occupational Surveys; Task Analysis; *Technical Reports	
IDENTIFIERS	Air Force; CODAP; Comprehensive Occupational Data Analysis Programs; UNIVAC 1108	

ABSTRACT

Documentation of the Univac 1108 Comprehensive Occupational Data Analysis Programs (CODAP) system is being published in a series of three technical reports covering the control card and programing aspects of the system, which is a highly interactive and efficient system of computer routines for analyzing, organizing, and reporting occupational information. CODAP is now capable of answering most of the wide variety of management questions which confront CODAP users. The document is the third report of the series, and it contains the source programs for the Univac 1108 version of CODAP. This includes 34 FORTRAN main programs, 78 FORTRAN and 20 Assembly Language subroutines. Additionally, a table of contents, an introduction, an entry point table, and an index are provided. A fourth report is in preparation by the Occupational Research Division of the Air Force Human Resources Laboratory (AFHRL), which covers the research and occupational applications of the CODAP system. (Author/AJ)

BEST COPY AVAILABLE

CE

AFHRL-TR-74-83

AIR FORCE HUMAN RESOURCES LABORATORY

CODAP:
SOURCE PROGRAM LISTINGS FOR THE UNIVAC 1108

By

Johnny J. Weissmuller, Sgt, USAF
Bruce B. Barton, A1C, USAF
C. R. Rogers

COMPUTATIONAL SCIENCES DIVISION
Lackland Air Force Base, Texas 78236

October 1974
Interim Report for Period January 1973 – September 1974

Approved for public release; distribution unlimited.

ED105199

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

CE003487

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235

NOTICE

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This interim report was submitted by Computational Sciences Division, Air Force Human Resources Laboratory, Lackland Air Force Base, Texas 78236, under project 7734, with Hq Air Force Human Resources Laboratory (AFSC), Brooks Air Force Base, Texas 78235. Dr. Raymond E. Christal, Chief of Occupational Research Division (OR), was project monitor, and Mr. William Phalen, OR, was the work unit scientist.

This report has been reviewed and cleared for open publication and/or public release by the appropriate Office of Information (OI) in accordance with AFR 190-17 and DoDD 5230.9. There is no objection to unlimited distribution of this report to the public at large, or by DDC to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved.

ROBERT A. BOTTENBERG, Chief
Computational Sciences Division

Approved for publication.

HAROLD E. FISCHER, Colonel, USAF
Commander

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFHRL-TR-74-83	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) CODAP: Source Program Listings for the Univac 1108	5. TYPE OF REPORT & PERIOD COVERED Interim Jan 73-Sep 74
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Johnny J. Weissmuller, Bruce B. Barton, C.R. Rogers	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computational Sciences Division, Air Force Human Resources Laboratory, Lackland Air Force Base, Texas 78236	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62703F 77340116
11. CONTROLLING OFFICE NAME AND ADDRESS HQ, Air Force Human Resources Laboratory (AFSC) Brooks Air Force Base, Texas 78235	12. REPORT DATE October 1974
	13. NUMBER OF PAGES 536
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Comprehensive Occupational Data Analysis Programs
CODAP
FORTRAN
Computer Programs
occupational survey
Source Programs
Subroutines
task analysis
work analysis
hierarchical clustering
job descriptions
job types

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Comprehensive Occupational Data Analysis Programs (CODAP) package is a highly interactive and efficient system of computer routines for analyzing, organizing, and reporting occupational information. Since its inception in 1960, CODAP has grown in tandem with advances in job analysis methodology and is now capable of answering most of the wide variety of management questions which confront CODAP users. This documentation of the UNIVAC 1108 CODAP system is being published in a series of 3 technical reports covering the (SEE REVERSE SIDE)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Block 20. ABSTRACT (Con't)

control card and programming aspects of the system. A fourth report is in preparation by the Occupational Research Division of AFHRL which covers the research and operational applications of the CODAP system. This document contains the source programs for the UNIVAC 1108 version of CODAP. This includes 34 FORTRAN main programs, 78 FORTRAN and 20 Assembly Language subroutines. Additionally, a table of contents, an introduction, an entry point table, and an index are provided.

PREFACE

The authors wish to express their gratitude to all those who have contributed to the preparation of this series of 3 technical reports:

AFHRL-TR-74-83 CODAP: Source Program Listings

AFHRL-TR-74-84 CODAP: Control Card Specifications

AFHRL-TR-74-85 CODAP: Programmer Notes

In particular, the following people have played a vital role in the development of these documents: Dr. Raymond E. Christal, Chief of the Occupational Research Division of AFHRL, Mr. William Phalen, also of that Division, and of course the secretaries, Betty L. Brooks and Laurel J. Betz, who not only typed but also helped in proofreading.

TABLE OF CONTENTS FOR PROGRAM ROSTER

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM		TITLE	PAGE
INTRO		INTRODUCTION TO CODAP	1
PHASES		PHASES OF A CODAP STUDY	4
ASFACT		ANALYSIS OF SECONDARY FACTORS	8
AUTOJT		AUTOMATIC JOB TYPE GENERATOR	17
AVALUE		AVERAGE VALUE PROGRAM	37
AVGPCT		AVERAGE BY PER CENT MEMBERS PERFORMING	46
CORREG		CORRELATIONS AND REGRESSIONS	62
	ALPHA	LOADER OF ALPHA CHARS SUBROUTINE (FROM PERSUB)	69
	CORRLB	CORRELATION SUBROUTINE (FROM PERSUB)	70
	CRTOTP	CORE TO TAPE SUBROUTINE (FROM PERSUB)	72
	PLEVEL	PROBABILITY LEVEL SUBROUTINE (FROM PERSUB)	73
	POSTAP	POSTION TAPE SUBROUTINE (FROM PERSUB)	75
	PRIMSC	PRINT MEANS,SD,&CORR SUBROUTINE (FROM PERSUB)	76
	PRINT1	PRINT SUBROUTINE (FROM PERSUB)	77
	REGREF	REGRESSION SUBROUTINE (FROM PERSUB)	79
	TAPGEN	TAPE GENERATION SUBROUTINE (FROM PERSUB)	86
	TPTOCR	TAPE TO CORE SUBROUTINE (FROM PERSUB)	87
	ZEROST	ZERO OUT ARRAY SUBROUTINE (FROM PERSUB)	88
DIAGRM		DIAGRAM OF THE CLUSTERED GROUPS	89
	STRT	SELECT STARTER GROUPS FOR THE DIAGRAM	95
	I2FA	CONVERT FLOATING POINT TO FIELDATA	98
DIST2X		TWO WAY DISTRIBUTION PROGRAM	99
	COMPA	COMPARE ALPHANUMERIC VALUES	129
	COMPC	COMPARE COLUMN VALUES IN PROPER MODE	130
	COMPN	COMPARE NUMERIC VALUES	131
	COMPRC	COMPARE ROW/COLUMN ORDERED PAIR VALUES	132
	DISTX1	SEARCH FOR PROPER INTERVAL IN THE DISTRIBUTION	133
	FORM2X	GENERATE FORMATS FOR DIST2X PRINTING PHASE	134
	OVRFLO	HANDLE DIST2X CORE TABLE OVERFLOW/ DISK MERGE	138
	SHIFT	SHIFT COLUMN TITLES IN DIST2X FOR RIGHT JUSTIFICATION	139
DUVARS		GENERATE AND REPORT DUTY VARIABLES	140
EXTRCT		EXTRACT REPORTS FROM THE REPORT FILES	150
	PRINTR	PRINT REPORTS EXTRACTED FROM REPORT FILE	154
	EDTPRT	EDIT AND PRINT REPORTS FROM REPORT FILE	156
	SETUPS	FILE ASSIGNMENT ROUTINE FOR EXTRCT	158
GRMBRS		GROUP MEMBERSHIP PROGRAM	161
GROUP		GROUP CLUSTERING PROGRAM	165

TABLE OF CONTENTS FOR PROGRAM ROSTER

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM	TITLE	PAGE
GRPDIF	GROUP DIFFERENCES PROGRAM	180
GRPSUM	GROUP SUMMARY PROGRAM	187
INPSTD	INPUT STANDARD PROGRAM	195
	RUNID RETURNS RUNID AND ACCOUNT CODE FOR CURRENT RUN	206
	ZERBLK CHANGE ZEROES TO BLANKS ON TASK/DUTY TITLE CARDS	207
JDINDX	INDEX A JOB DESCRIPTION FILE	208
JOBGRP	JOB DESCRIPTIONS OF GROUPS AT GIVEN STAGES	209
JOBIND	JOB DESCRIPTIONS FOR INDIVIDUALS	217
	REPENI CLOSE REPORTS & PUNCH SPECIAL JOBIND REQUEST CARD	234
JOBINV	JOB INVENTORY FOR TASKS AND DUTIES	235
JOBSPC	JOB DESCRIPTIONS OF SPECIAL GROUPS	239
KPATH	RE-ORDER HISTORY FILE INTO KPATH FILE	253
MTXPRT	PRINT OVERLAP MATRIX OF GIVEN GROUPS	256
	OVLAP COMPUTE OVERLAP OF GIVEN GROUPS	259
	DISKLD LOAD DISK FOR MTXPRT PROGRAM	262
OVRLAP	OVERLAP TASK DATA AND CREATE MATRIX	264
	RFILE RANDOM ACCESS CASE FILE OF NON-ZERO TASK RESPONSES	274
PRDICT	PRINT THE VARIABLE DICTIONARY	282
PRIJOB	SELECT PRIMARY JOB IDENTIFIERS	283
PROGEN	GENERATE A FORTRAN PROGRAM FOR CODAP USAGES	293
	SISO SYSTEM FILE/ELEMENT INTERFACE ROUTINE	309
PRTVAR	PRINT INFORMATION FROM A HISTORY/KPATH FILE W/SORT	312
REXALL	INTER-RATER RELIABILITY W/ALL OPTIONS	317
SETCHK	CHECK CASE DATA SETUP	333
TSETUP	SETUP JOB DESCRIPTION AND/OR REPORT FILES	342
TSKGRP	TASK INDEX PROGRAM ON RESTRICTED GROUP OF TASKS	344
TSKNDX	TASK INDEX PROGRAM	357
VARGEN	VARIABLE GENERATING PROGRAM	369
	APLWTS APPLY WEIGHTS TO RATINGS	372
	DICTRW DICTIONARY READ/WRITE SUBROUTINE	375

TABLE OF CONTENTS FOR PROGRAM ROSTER

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM	TITLE		PAGE
VARMEN	VARIABLE MEANS & STANDARD DEVIATIONS (CORREG W/O MATRIX)		377
VARSUM	SUMMARY OF DISTRIBUTION FOR VARIABLE VALUES		384
SUBLIB	SUBROUTINE LIBRARY		400
ENTRY	ENTRY POINT TABLE		400
	ASGA	ASSIGN AN ALREADY CATALOGUED FILE	403
	ASGAM	ASSIGN A CYCLE OF A 3-REEL FILE	405
	ASGC	LOAD A BLANK TAPE TO BE WRITTEN ON & CATALOGUED	407
	ASGCM	LOAD BLANK TAPES TO CREATE 3-REEL "M" FILE	408
	AUTORV	AUTOMATIC FILE REVERSION: RECOVER FROM CYCLE ERROR	410
	BLANK	BLANK OUT AN ARRAY (NWORD,IARRAY)	411
	CFHIO	COBOL FILE HANDLER SUBROUTINE	412
	CPI	PACKS FORMAT CHARACTERS INTO A SINGLE WORD	421
	CYCLES	HANDLE "M" FILE CYCLE ASSIGNMENTS	423
	DATACV	CONVERT FIELDATA TO BINARY, BINARY TO FIELDATA	425
	DATETM	RETURN DATE AND TIME IN NEAT FORM	427
	DELAY	CAUSE PROGRAM TO WAIT SPECIFIED # OF SECONDS	428
	DISK	SIMULATION OF IBM 7040 RANDOM ACCESS ROUTINES	429
	DKSTAT	SEE IF A MASS STORAGE FILE IS ASSIGNED	430
	DREAD	READ A FORMATTED-FORTRAN WRITTEN MASS STORAGE FILE	431
	ERII	PROVIDE OPERATOR INTERROGATION OF PROGRAM ID/STATUS	436
	FACREJ	DECODE OCTAL FACILITY REJECTION CODE TO ENGLISH	438
	FMTGEN	GENERATE FORTRAN FORMATS FROM CODAP FORMAT CARDS	441
	FORMAT	EXTRACT VARIABLE'S FORMAT FROM VARIABLE DICTIONARY	446
	FREDEV	FREE DEVICE: MAKE FORTRAN FORGET DEVICE WAS ASSIGNED	448
	FREE	FREE FILE AND ASSOCIATED DEVICE	449
	FREES	FREE FILE BUT RETAIN TAPE DRIVE FOR ANOTHER ASSIGNMENT	450
	FSORT	FORTRAN SORT SUBROUTINE / SYSTEM INTERFACE	451
	GETFIL	ISSUE ASSIGN FOR A TAPE UNTIL A DRIVE IS AVAILABLE	452
	GETMAS	ISSUE ASSIGN FOR MASS STORAGE UNTIL ACCEPTED	453
	GETPCD	RETREIVE INFORMATION FROM THE PROGRAM CONTROL TABLE (DEC)	454
	GETPCT	RETREIVE INFORMATION FROM THE PROGRAM CONTROL TABLE (OCT)	455
	H2ACND	SORT INTO ASCENDING ORDER ON H2 OF WORD	456
	H2SORT	SORT INTO DESCENDING ORDER ON H2 OF WORD	457
	INSERT	CHARACTER INSERTION SUBROUTINE	458
	IRANF	RANDOM NUMBER GENERATOR & INITIALIZATION ROUTINE	459
	JDCOPY	COPIES JOB DESCRIPTION FILE	460
	JDEOF	FINDS THE END-OF-FILE ON JOB DESCRIPTION FILE	461
	LASCMP	LOGICAL ALPHANUMERIC STRING COMPARISON	462
	LINK	FORTRAN SORT PACKAGE LINKAGE	463
	LISORT	POINTER LIST SORT (ASCENDING INTEGER)	467
	LSORT	POINTER LIST SORT (DESCENDING FLOATING POINT)	468
	MSKOP1	PERFORM MASK OPERATIONS ON VARIABLE MASK ARRAYS	469
	MSKOP2	PERFORM MASK OPERATIONS ON A SINGLE FIXED MASK	470
	NEXREL	FREE TAPE FILE ON A DRIVE, THEN ASSIGN NEW FILE TO IT	471
	NOHEAD	CHANGE SYSTEM DEFAULTS ON PAGE OVERFLOWS	472
	NRAND	RANDOM NUMBER GENERATOR	473
	OMSG	OPERATOR MESSAGE/INTERACTION SUBROUTINE (ON "LOAD" CONSOLE)	474

TABLE OF CONTENTS FOR PROGRAM ROSTER

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM		TITLE	PAGE
	OMSGO	OPERATOR MESSAGE/INTERACTION SUBROUTINE (ON "TD" CONSOLE)	476
	PRTDIC	PRINT THE VARIABLE DICTIONARY (PRDICT,VARGEN,PROGEN)	478
	RANDOM	STANDARD RANDOM SELECTION SUBROUTINE (IRAND,SRAND)	479
	RANF	FLOATING POINT RANDOM NUMBER FUNCTION (MUST CALL IRANF)	480
	RANSEQ	RANDOMLY SEQUENCE AN ARRAY	481
	REPEND	CLOSE REPORTS AND PUNCH A REQUEST CARD	482
	RESDRV	RESERVE A TAPE DRIVE (BEING PHASED OUT: USE FREES)	483
	ROSTER	CONTROL ROSTER & ERROR MESSAGES (PRINTC,ERRMSG,RESET)	484
	RPCOPY	COPIES A REPORT FILE FOR A NEW CYCLE	485
	RPEOF	LOCATES AN END-OF-FILE ON A REPORT FILE	486
	RPINDX	PRINTS AN INDEX OF A REPORT FILE	487
	SAMSEL	NEW SAMPLE SELECTION FOR JOBSPC,AVALUE,AVGPCT,TSKNDX	488
	SETUP	THE STANDARD FILE SETUP SUBROUTINE	491
	SETUP9	FILE ASSIGNMENT ROUTINE FOR JOBSPC & JOBGRP	494
	STATS	STATISTICS GATHERER: PRIMARILY FOR INPSTD	497
	STATUS	CHECK IF A GIVEN TAPE FILE IS ALREADY ASSIGNED	498
	SUBRTN	DICK ROGERS' UTILITY SUBROUTINE PACKAGE	499
	TABL3	FORTRAN SORT PACKAGE	515
	TMTCSC	MOVE CONTENTS OF ARRAYS FROM ONE TO ANOTHER	516
	TREAD	READ FORMATTED-FORTRAN WRITTEN TAPE FILES	517
	ZERO	STORE FLOATING POINT ZEROES INTO A SPECIFIED ARRAY	522
	ZEROI	STORE INTEGER ZEROES INTO A SPECIFIED ARRAY	523
INDEX		INDEX FILE USED AS INPUT TO PROGRAM ROSTER ROUTINE	524

THE COMPREHENSIVE OCCUPATIONAL DATA ANALYSIS PROGRAMS (CODAP) SYSTEM HAS BEEN UNDER CONTINUOUS DEVELOPMENT FOR THE PAST FIFTEEN YEARS. IN ITS CURRENT FORM, IT IS A JOINT PRODUCT OF THE COMPUTATIONAL SCIENCES DIVISION AND THE OCCUPATIONAL RESEARCH DIVISION OF AFHRL. IN GENERAL, THE OCCUPATIONAL RESEARCH DIVISION HAS DEVELOPED INPUT-OUTPUT SPECIFICATIONS FOR PROGRAMS AND PROGRAM MODIFICATIONS, WHILE THE COMPUTATIONAL SCIENCES DIVISION HAS PROVIDED PROGRAMMING SERVICES. HOWEVER, EVEN THIS DISTINCTION HAS NOT ALWAYS BEEN A CLEAR-CUT ONE, SINCE SUGGESTIONS FOR PROGRAM IMPROVEMENTS HAVE COME FROM INDIVIDUALS IN BOTH DIVISIONS.

SINCE ITS CONCEPTION, DEVELOPMENT OF THE CODAP SYSTEM HAS BEEN UNDER THE GENERAL DIRECTION OF DR. RAYMOND E. CHRISTAL, CHIEF OF THE OCCUPATIONAL RESEARCH DIVISION. DR. CHRISTAL ALSO PERSONALLY DESIGNED MANY OF THE PROGRAMS AND PROGRAM MODIFICATIONS.

THE EARLIEST CODAP PROGRAMS WERE WRITTEN BY MR. DANIEL RIGNEY AND MR. WAYNE E. FISHER. THIS VERSION WAS LATER EXPANDED AND REPROGRAMMED (UNDER CONTRACT) FOR EXECUTION ON AN IBM 7040 BY MR. IRWIN R. OATS AND MR. JAMES R. MILLS OF THE COMPUTER SCIENCES CORPORATION AND LATER OF OATS-MILLS INCORPORATED, HOUSTON, TEXAS. (MR. OATS AND MR. MILLS HAVE TRANSLATED CODAP PROGRAMS FOR EXECUTION ON THE IBM 360 AND 370 SERIES COMPUTERS, UNDER CONTRACT WITH THE U.S. MARINE CORPS AND THE DEPARTMENT OF DEFENSE.)

SINCE 1970 ALL CODAP INNOVATIONS AND MODIFICATIONS HAVE BEEN PROGRAMMED IN-HOUSE BY THE COMPUTATIONAL SCIENCES DIVISION. DURING THIS PERIOD, MANY NEW PROGRAMS WERE ADDED TO THE SYSTEM, AND NEARLY ALL OLD PROGRAMS WERE EXTENSIVELY MODIFIED. DURING THE LAST TWO YEARS THE CODAP SYSTEM HAS BEEN COMPLETELY REWRITTEN FOR EXECUTION ON THE UNIVAC 1108. THIS REPROGRAMMING WAS FIRST UNDERTAKEN BY MSGT. WILLIAM D. STACEY, WITH THE ASSISTANCE OF SGT. JOHNNY J. WEISSMULLER. AFTER MSGT. STACEY'S DEPARTURE FROM AFHRL, THE TRANSLATION AND REPROGRAMMING WERE COMPLETED AND CHECKED OUT BY SGT. WEISSMULLER, ASSISTED BY AIC BRUCE B. BARTON. IT IS THIS VERSION OF THE CODAP SYSTEM WHICH IS DOCUMENTED IN THIS TECHNICAL REPORT.

MANY INDIVIDUALS HAVE CONTRIBUTED TO THE CODAP SYSTEM, AND IT WOULD BE DIFFICULT TO SPECIFICALLY ENUMERATE EACH CONTRIBUTION. HOWEVER, THE AUTHORS WOULD BE REMISS NOT TO MENTION THOSE INDIVIDUALS WHO HAVE PARTICIPATED IN THE MAINSTREAM EFFORT. IT IS POSSIBLE, WITH THE PASSING OF TIME AND THE FAILING OF MEMORIES, THAT THE FOLLOWING HISTORY CONTAINS SERIOUS OMISSIONS AND INACCURACIES, BUT IT IS CORRECT IN TERMS OF CURRENTLY AVAILABLE INFORMATION.

THE HIERARCHICAL CLUSTERING PROGRAMS, OVRLAP,GROUP,AND KPATH, WERE

INITIALLY DESIGNED BY JOE H. WARD, JR. AND R. E. CHRISTAL, WITH ORIGINAL PROGRAMMING BY DANIEL RIGNEY AND WAYNE E. FISHER. THE REMAINDER OF THE FIRST CODAP PACKAGE WAS DESIGNED BY R. E. CHRISTAL, WITH NUMEROUS SUGGESTIONS FROM I. R. OATS, J. R. HILLS AND OTHERS. THIS PIONEERING EFFORT INCLUDED JOBGRP, JOBSPC, ASFACT, PRDICT, GRMBRS, PRTVAR, VARSUM, GRPSUM, GRPDIF, MXTPRT, TSETUP, INPSTD, VARGEN, EXTRCT, AND JOBINV. THE PROGRAMMING FOR THESE WAS ACCOMPLISHED BY I. R. OATS, J. R. HILLS, AND D. W. HARTMANN.

THE TABLE BELOW LISTS THE PROGRAMS DEVELOPED AFTER THE INITIAL PACKAGE:

PROGRAM	INITIAL DESIGN	INITIAL PROGRAMMING
SETCHK	W. L. WELLS	EDWARD L. ADAMS
PROGEN	EDWARD L. ADAMS	EDWARD L. ADAMS
TSKNDX	R. E. CHRISTAL	WILLIAM D. STACEY
AVALUE	R. E. CHRISTAL	WILLIAM D, STACEY
DIST2X	C. R. ROGERS	JOHNNY J. WEISSMULLER
CORREG	JOE H. WARD, JR. ROBERT A. BOTTENBERG	JANICE BUCHHORN KATHLEEN HALL WILLIAM D. STACEY
AUTOJT	WILLIAM PHALEN	PAUL ARON
PRIJOB	WILLIAM PHALEN R. E. CHRISTAL	COMPUTER SCIENCES CORPORATION
DUVARS	WILLIAM PHALEN	WILLIAM D. STACEY
REXALL	WILLIAM PHALEN JOHNNY J. WEISSMULLER R. E. CHRISTAL	JOHNNY J. WEISSMULLER
AVGPCT	WILLIAM PHALEN	JOHNNY J. WEISSMULLER
JDINDX	JOHNNY J. WEISSMULLER	JOHNNY J. WEISSMULLER
JOBIND	R. E. CHRISTAL	EDWARD L. ADAMS
TSKGRP	HARRY LAWRENCE JOHNNY J. WEISSMULLER	JOHNNY J. WEISSMULLER
DIAGRM	WILLIAM PHALEN R. E. CHRISTAL PHILIP AITKEN-CADE	COMPUTER SCIENCES CORPORATION

AMONG OTHER INDIVIDUALS WHO HAVE PLAYED A MEANINGFUL PART IN THE DEVELOPMENT, IMPROVEMENT AND MAINTENANCE OF THE CODAP SYSTEM, ARE THE FOLLOWING:

DR. ROBERT A. BOTTENBERG MR. JIM SOUTER MR. C. R. ROGERS	MONITORING AND DIRECTION OF PROGRAMMING
MR. J. MYER	SYSTEMS AND PROGRAMMING
JOE MORSH WAYNE ARCHER	NUMEROUS SUGGESTIONS FOR PROGRAM CHANGES
HARRY KUDRICK W. L. WELLS ROBERT L. VANCE BILL FELTNER, JR. WESLEY C. HILL STEVE D. POOLE TERRY D. LEWIS	PROGRAM EXECUTION AND SUGGESTIONS FOR IMPROVEMENTS

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

INTRODUCTION

I. DATA PREPARATION

A. GENERALLY THIS STEP INCLUDES UTILITY PROGRAMS FOR SORTS, MERGES, ETC. MAY INCLUDE OPTICAL SCANNING AND REFORMATTING ROUTINES.

B. FINAL RUN WILL BE A "SETCHK" RUN TO PRODUCE EITHER THE RAW-DATA FILE FOR "INPSTD", OR A RATER FILE FOR EITHER "REXALL" OR "TSKNDX".

II. CODAP FILE CREATION

A. "INPSTD" CREATES THE HISTORY DATA FILE FROM THE RAW-DATA FILE AND THE CONTROL CARDS. CASES ARE SEQUENCED THE SAME AS ON THE RAW-DATA FILE AND SOME VARIABLES (C003-6,C009) ARE LEFT ZERO, TO BE FILLED IN BY "KPATH".

B. "TSETUP" INITIALIZES THE JOB DESCRIPTION AND/OR REPORT FILES. EACH OF THESE FILES CONSISTS OF 3 REELS OF TAPE. EACH REEL IS A COPY OF THE FILE, AND EACH COPY IS CALLED A FILE CYCLE. CYCLES ARE DENOTED (0), (-1), AND (-2). CYCLE (0) IS THE MOST CURRENT COPY AND CYCLES (-1) AND (-2) ARE THE FIRST AND SECOND BACKUP, RESPECTIVELY. ONLY "JOBGRP" AND "JOBSPC" ARE ABLE TO ADD RECORDS TO THE JOB DESCRIPTION FILE, WHILE MANY PROGRAMS MAY ADD REPORTS TO THE REPORT FILE. (SEE SECTION VI B).

III. HIERARCHICAL CLUSTERING (OPTIONAL)

A. "OVRLAP" COMPUTES THE SIMILARITY BETWEEN ALL PAIRS OF CASES AND CREATES THE OVRLAP MATRIX FILE.

B. "GROUP" USES THE OVRLAP MATRIX FILE TO HIERARCHICALLY CLUSTER CASES INTO LARGER AND LARGER GROUPS. EACH MERGER IS CALLED A STAGE AND EACH STAGE UNIQUELY DEFINES A GROUP OF INDIVIDUALS WHICH WAS CREATED AT THAT STAGE. THE OUTPUT FILE FROM "GROUP" IS CALLED THE STAGE DATA FILE.

C. "KPATH" USES THE INFORMATION ON THE STAGE DATA FILE TO REORDER THE CASES ON THE HISTORY DATA FILE. THE HISTORY DATA FILE IS REORDERED, ADDITIONAL INFORMATION IS INSERTED INTO EACH CASE RECORD (C003-6,C009) AND THE KPATH FILE IS WRITTEN OUT. THE KPATH FILE MAY BE USED BY ANY PROGRAM WHICH COULD USE THE HISTORY DATA FILE, BUT THE CONVERSE IS NOT NECESSARILY TRUE. (SEE "JOBGRP" IN SECTION V B)

D. "GRMBRS" READS THE KPATH FILE, PRODUCES A REPORT ON THE GROUP MERGERS, AND CREATES THE GRMBRS FILE TO BE USED BY "DIAGRM".

E. "DIAGRM" READS THE GRMBRS FILE AND PRINTS A PICTORIAL REPRESENTATION OF THE HIERARCHICAL GROUPING PROCESS.

IV. FILE MODIFICATION

A. THESE PROGRAMS MAY BE EXECUTED AT ANY TIME AFTER "INPSTD" OR "KPATH" (DEPENDING ON THE FILE USED.) WHEN AND IF THEY ARE USED IS PURELY AT THE DISCRETION OF THE TASK SCIENTIST OR PROJECT DIRECTOR.

B. "DUVARS" CREATES A NEW HISTORY/KPATH FILE WITH ADDITIONAL BACKGROUND (VXXX) VARIABLES BASED UPON DUTY PERFORMANCE.

C. "VARGEN" CREATES A NEW HISTORY/KPATH FILE WITH ADDITONAL COMPUTED (CXXX) VARIABLES BASED UPON OPTIONS USING A COMBINATION OF A WEIGHT DECK AND THE INDIVIDUAL'S TASK VALUES.

D. "PROGEN" IS A FORTRAN PROGRAM GENERATOR. IT GENERATES A FORTRAN SOURCE DECK WHICH WILL HANDLE THE OVERHEAD REGIONS OF A HISTORY/KPATH FILE, ALLOWING THE EXPERT CODAP USER TO CREATE, MODIFY, OR USE EITHER BACKGROUND OR COMPUTED VARIABLES. NOTE: "PROGEN" MERELY CREATES A SOURCE DECK. IT IS LEFT TO THE USER TO PRODUCE AN EXECUTABLE PROGRAM FROM THAT DECK.

V. GROUP IDENTIFICATION

A. "JOBSPC" WILL IDENTIFY ALL INDIVIDUALS WITH SPECIFIED CHARACTERISTICS BASED UPON THE VALUES OF EITHER BACKGROUND OR COMPUTED VARIABLES. EITHER A HISTORY OR KPATH FILE MAY BE USED, BUT IF HIERARCHICAL CLUSTERING IS TO BE DONE AT ALL, IT IS RECOMMENDED THAT ONLY THE KPATH FILE BE USED. OPTIONALLY, THE REPORT MAY BE COPIED TO THE REPORT FILE AND THIS GROUP RECORDED BY ADDING AN ENTRY ON THE JOB DESCRIPTION FILE. THIS ALLOWS THE GROUP TO BE IDENTIFIED LATER BY MERELY INSERTING THE REQUEST CARD WHICH WILL BE PUNCHED.

B. "JOBGRP" WILL IDENTIFY ALL INDIVIDUALS WHO WERE IN THE GROUP THAT WAS FORMED AT A SPECIFIED STAGE OF THE GROUPING PROCESS. NOTE THAT ONLY THE KPATH FILE MAY BE USED AS INPUT AS THE CASES ARE NOT PROPERLY SEQUENCED ON THE HISTORY FILE, AND REQUISITE INFORMATION WOULD NOT BE AVAILABLE. ALSO, THE FUNCTION OF "JOBGRP" MAY BE ABSORBED BY "JOBSPC" INASMUCH AS SPECIFYING A RANGE ON C009 (KPATH SEQUENCE NUMBER) CAN IDENTIFY A MERGING GROUP AND THE RANGE MAY BE FOUND ON EITHER THE "GRMBRS" OR "DIAGRM" OUTPUTS. "JOBGRP", LIKE "JOBSPC", MAY OPTIONALLY ADD REPORTS TO THE REPORT FILE AND MAKE ENTRIES ON THE JOB DESCRIPTION FILE.

VI. REPORT PROGRAMS

NOTE: A PRECEDING ASTERISK (*) DENOTES THE PROGRAM REQUIRES THE JOB DESCRIPTION FILE.

A. INTERMEDIATE REPORTS: CANNOT BE ADDED TO REPORT FILE.

1. "CORREG" COMPUTES CORRELATIONS AND REGRESSIONS ON UP TO 100 COMPUTED OR BACKGROUND VARIABLES.

* 2. "AUTOJT" PRODUCES MEASURES OF GROUP DIFFERENCES ON UP TO 850 PAIRS OF GROUPS.

3. "DIST2X" PRINTS A TWO-WAY DISTRIBUTION ON SPECIFIED PAIRS OF COMPUTED OR BACKGROUND VARIABLES.

* 4. "JDINDX" LISTS AN INDEX OF ALL ENTRIES ON A JOB DESCRIPTION FILE.

5. "VARMEN" COMPUTES MEANS AND STANDARD DEVIATIONS ON UP TO 100 COMPUTED OR BACKGROUND VARIABLES. (REDUCED VERSION OF "CORREG")

B. INTERMEDIATE OR FINAL REPORTS: MAY BE ADDED TO REPORT FILE.

* 1. "ASFACT" DISTRIBUTES TASK RATINGS ON SECONDARY FACTORS.

2. "AVALUE" COMPUTES THE MEAN OF A GIVEN VARIABLE FOR EACH TASK BASED ON THE CASES WHICH PERFORMED THAT TASK.

3. "AVGPCT" IS THE SAME AS "AVALUE" EXCEPT THAT THE PERCENT OF A GROUP PERFORMING IS USED, RATHER THAN THE ACTUAL NUMBER OF CASES. THIS WILL TEND TO REDUCE SAMPLING BIAS.

* 4. "GRPDIF" PRINTS A TASK-BY-TASK DIFFERENCE BETWEEN GROUPS.

* 5. "GRPSUM" PRINTS TASK SUMMARIES OF SEVERAL GROUPS PER PAGE.

* 6. "JOBIND" PRINTS JOB DESCRIPTIONS FOR EACH CASE IN A GIVEN GROUP.

7. "JOBINV" PRINTS ALL THE TASK AND DUTY STATEMENTS IN THE INVENTORY.

* 8. "MTXPRT" PRINTS AN OVERLAP MATRIX FOR SPECIFIED GROUPS.

9. "PRDICT" PRINTS THE VARIABLE DICTIONARY WHICH IDENTIFIES ALL THE COMPUTED (CXXX) AND BACKGROUND (VXXX) VARIABLES.

* 10. "PRIJOB" PRINTS THE PRIMARY JOB IDENTIFIERS SUCH AS THE MOST PERFORMED OR MOST TIME CONSUMING TASKS.

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

- 11."PRTVAR" SORTS AND PRINTS VARIABLES FOR ALL INDIVIDUALS IN SPECIFIED GROUPS.

12."REXALL" COMPUTES THE INTER-RATER RELIABILITY COEFFICIENT FROM THE RATER DATA FILE, AND OPTIONALLY PUNCHES A WEIGHT DECK.

13."TSKGRP" REPORTS THE TASK INDEX VALUES FROM A RATER DATA FILE OR A WEIGHT DECK FOR A SPECIFIED SET OF TASKS AND A SPECIFIED GROUP OF CASES.

14."TSKNDX" REPORTS THE TASK INDEX VALUES FROM A RATER DATA FILE OR A WEIGHT DECK FOR A SPECIFIED GROUP OF CASES.

- 15."VARSUM" WILL DISTRIBUTE THE VALUES OF SPECIFIED VARIABLES FOR SPECIFIED GROUPS.

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      DIMENSION JA(50),IREL(1000),ITKDY(30),IC(1),C(500),
     1 IV(1000),MSK(200,13),MBUF(14),INF(1600),IBUF(11),
     2 ICORE(10,50,12), LGRP(4,12), NEED(200),AK(1750),
     3 JDA(50),JDMSK(600),ITL(8),IHEAD(14,6),JSEQGP(12,2),
     4 KTASK(1700),LTASK(69),KOTH(12),GPM(12),GPS(12),LATASK(26),IDG(12)
C DEFINITIONS:
C
C     ICORE: FIRST SUBSCRIPT = TASK RATING (0 TO 9, BIASED+1)
C            SECOND SUBSCRIPT = TASK NUMBER RELATIVE TO THE "PAGE"
C                               OF THE 50 TASKS CURRENTLY IN CORE
C            THIRD SUBSCRIPT = GROUP NUMBER (1 TO NGRPSLMAX=12J)
C
C     LGRP: FIRST SUBSCRIPT = 1 FOR GROUP ID (A6)/ SUM OF N*IVAL (I6)
C                             2 FOR MATRIX ID (A6)/ SUM OF N*IVAL**2
C                             3 FOR MEMBERS ON PRIMARY FILE (I6)/ SUM N
C                             4 FOR MEMBERS ON SECONDARY FILE (I6)
C           SECOND SUBSCRIPT = GROUP NUMBER (1 TO NGRPSLMAX=12J)
C
C     MSK: MASK ARRAYS FOR ALL GROUPS. MSK(XXX,13) IS RESERVED
C          FOR A COMPOSITE MASK = LOGICAL "OR" OF ALL INDIVIDUAL
C          GROUP MASKS.  IT IS EQUIVALENCED TO "NEED" FOR QUICK
C          ACCESS AND NOTATION.
C
C     JSEQGP: FIRST SUBSCRIPT = GROUP NUMBER
C             SECOND SUBSCRIPT = 1 FOR JSEQ NUMBER
C                                2 FOR GROUP NUMBER
C
C
      EQUIVALENCE (MSK,JA,IREL,IC,C),(IV,C(2)),(AK,JDMSK,INF),
     1 (MSK(1,13),NEED),(MSK,KOTH)
      DATA IBUF/'11',6*' ',751,14,'IDSRRP',' '/
      DATA LIMIT/O377777777777/
      INTEGER V001
      INM = 10
      INS = 11
      IJD = 27
      IRP = 26
      ISI = 20
      ISO = 21
      IDV = 22
      IDC = 23
      CALL SETUP(10,11,27,26,0,0,20,21,22,23,0,JSEQ)
      CALL RESET
      READ(5,800) ID1,ID2,IDEBUG
  800 FORMAT(7X,A6,A2,1X,A1)
      CALL PRINTC
      READ(INM) N,(JA(I),I=1,N)
      NCASE = JA(3)
      NDICT = JA(4)
      MATRIX = JA(6)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
          NTASK = JA(12)
          READ(INH) NV,(IREL(I),I=1,NV),NDUTY,(ITKDY(I),I=1,NDUTY)
          DO 1 IDICT = 1, NDICT
          READ(INH)
    1     CONTINUE
          DO 2 ICASE = 1, NCASE
          READ(INH) N,(C(I),I=1,N),NV,(IV(I),I=1,NV)
          WRITE(IDV) IV(1)
          WRITE(IDC) IC(1)
    2     CONTINUE
          ENDFILE IDV
          ENDFILE IDC
          CALL FREE(INH)
          CALL FSORT(IDV,1,0,0)
    3     READ(5,801,END=999) IDREP,ITL,NHEAD,ISTART,MIN,MAX,IREP,IFREQ,
         1 IOTH,ISTAT
  801     FORMAT(A6,1X,8A6,1X,A1,1X,A2,6(1X,A1))
          CALL PRINTC
          IF(MIN.EQ.' ') MIN = '1'
          IF(MAX.EQ.' ') MAX = '7'
          CALL BCDBIN(NHEAD,NHEAD,1,$900)
          CALL BCDBIN(MIN,MIN,1,$900)
          CALL BCDBIN(MAX,MAX,1,$900)
          CALL BCDBIN(IREP,IREP,1,$900)
          CALL BCDBIN(IFREQ,IFREQ,1,$900)
          CALL BCDBIN(IOTH,IOTH,1,$900)
          CALL BCDBIN(ISTAT,ISTAT,1,$900)
          NVALS = (MAX-MIN) + 1
          IF(NHEAD.EQ.0) GO TO 30
          DO 25 J = 1, NHEAD
          READ(5,805) (IHEAD(I,J),I=1,14)
          CALL PRINTC
   25     CONTINUE
   30     CONTINUE
  805     FORMAT(13A6,A2)
          IGRP = 0
    5     READ(5,802,END=6) IEND, LATASK, LSEQ
  802     FORMAT(A3,1X,26R1,45X,A5)
          CALL PRINTC
          IF(IEND.EQ.'END') GO TO 6
          CALL BCDBIN(LSEQ,LSEQ,5,$901)
          IGRP = IGRP + 1
          JSEQGP(IGRP,2) = IGRP
          JSEQGP(IGRP,1) = LSEQ
          GO TO 5
  901     CONTINUE
          WRITE(6,803)
  803     FORMAT(' INVALID REPORT NUMBER ABOVE - REQUEST IGNORED.')
          GO TO 5
   13     CONTINUE
```

LINE # TEXT...

```
          NC35 = (NCASE+34)/35
          DO 14 I = 1,NC35
          MSK(I,1) = LIMIT
          NEED(I) = LIMIT
   14     CONTINUE
          NGRPS = 1
              LGRP(1,1) = '  FILE'
              LGRP(2,1) = ' '
              LGRP(3,1) = NCASE
          LGRP(4,1) = 0
          GO TO 15
    6     CONTINUE
          REWIND IDV
          REWIND IDC
          NGRPS = IGRP
          IF(NGRPS.EQ.0) GO TO 13
          CALL ZERMSK(NCASE,NEED,LEN)
          CALL LISORT(NGRPS,JSEQGP(1,1),JSEQGP(1,2))
          REWIND IJD
          DO 10 JD = 1, NGRPS
          JISEQ = JSEQGP(JD,2)
          ISEQ = JSEQGP(JISEQ,1)
    7     CONTINUE
          READ(IJD,END=903) IDSP,NTD,(AK(I),I=1,NTD),NC35,(JDMSK(I),I=1,
         1 NC35),JSEQ,NC,(JDA(I),I=1,NC)
          IF(JSEQ.LT.ISEQ) GO TO 7
          IF(JSEQ.NE.ISEQ) GO TO 904
          IF(JDA(3).NE.NCASE) GO TO 906
          CALL ZERMSK(NCASE,MSK(1,JISEQ),LEN)
          ITYPE = 1
          IF(JDA(6).EQ.' '.OR.JDA(6).EQ.0) ITYPE = 0
          REWIND IDC
          DO 8 ICASE = 1, NCASE
          READ(IDC) KPATH
          CALL CHKMSK(ICASE,JDMSK,$8)
          IF(ITYPE.EQ.0) KPATH = ICASE
          CALL SETMSK(KPATH,MSK(1,JISEQ))
          CALL SETMSK(KPATH,NEED)
          IF(IDEBUG.EQ.'1') WRITE(6,850) ICASE
  850     FORMAT(' SELECTED CASE ',I6)
    8     CONTINUE
          IDG(JISEQ) = IDSP
          LGRP(1,JISEQ) = IDSP
          LGRP(2,JISEQ) = JDA(6)
          LGRP(3,JISEQ) = JDA(17
          LGRP(4,JISEQ) = 0
          IF(NC35.GT.200) GO TO 90:
   10     CONTINUE
   15     CONTINUE
          DO 11 I = 1, 11
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
         INF(I) = IBUF(I)
  11     CONTINUE
  C ***** OPEN RAW DATA INPUT FILE ********
         CALL OPENF(INF)
         LIN = 20
         LOU = 21
         NPAGE = (NTASK+49)/50
         CALL ZEROI(ICORE,6000)
         REWIND LIN
         REWIND LOU
         DO 12 IPAGE = 1, NPAGE
         WRITE(LIN) (((ICORE(I,J,K),I=1,NVALS),J=1,50),K=1,NGRPS)
  12     CONTINUE
         ENDFILE LIN
         REWIND LIN
         CALL READF(INF,MBUF,$999)
         DECODE(804,MBUF) V001,LTASK,IDCARD
  804    FORMAT(A4,69A1,A2)
  C   **********  BEGIN MAIN DISTRIBUTION LOOP ***********
         DO 500 ICASE = 1, NCASE
         READ(IDV) IDCASE
         CALL CHKMSK(ICASE,NEED,$500)
         IF(IDEBUG.EQ.'1') WRITE(6,850) ICASE
  100    IF(IDCASE.EQ.V001) GO TO 200
         IF(IDEBUG.EQ.'1') WRITE(6,851) IDCASE, V001
  851        FORMAT(' IDCASE = ',A6,'  V001 = ',A6)
         IF(FLD(0,24,IDCASE).LT.FLD(0,24,V001)) GO TO 500
         CALL READF(INF,MBUF,$501)
         DECODE(804,MBUF) V001,LTASK,IDCARD
         GO TO 100
  C  ********* FOUND MATCHING SECONDARY CASE ***********
  200    CONTINUE
         IF(IDCARD.EQ.ISTART) GO TO 201
         CALL READF(INF,MBUF,$501)
         DECODE(804,MBUF) V001,LTASK,IDCARD
         IF(IDCASE.NE.V001) GO TO 909
         IF(IDEBUG.EQ.'1') WRITE(6,852) ISTART,IDCARD
  852        FORMAT(' ISTART = ',A6,'  IDCARD = ',A6)
         GO TO 200
  C ********** SET UP AND BIAS TASK RESPONSES **********
  201    IST = 0
         CALL ZEROI(KTASK,NTASK)
  202    CONTINUE
         DO 205 INDEX = 1, 69
         ITASK = IST + INDEX
         IF(ITASK.GT.NTASK) GO TO 206
         CALL BCDBIN(LTASK(INDEX),IVAL,1,$205)
         IF(IVAL.LT.MIN.OR.IVAL.GT.MAX) GO TO 205
         KTASK(ITASK) = IVAL + 1 - MIN
  205    CONTINUE
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      ITEST = IST + 69
      IF(ITEST.EQ.NTASK) GO TO 206
      IST = IST + 69
      GO TO 202
206   CONTINUE
C
C  ••••••••••••• BEGIN "PAGING" TASKS ••••••••••••••
C
      DO 400 IPAGE = 1, NPAGE
      READ(LIN) (((ICORE(I,J,K),I=1,NVALS),J=1,50),K=1,NGRPS)
      DO 300 IGRP = 1, NGRPS
      CALL CHKMSK(ICASE,MSK(1,IGRP),$300)
      IF(IDEBUG.EQ.'1') WRITE(5,853) ICASE,IGRP
853      FORMAT(' USED CASE ',I6,' FROM GROUP #',I6)
      JTPAGE = IABS(50*(IPAGE-1))
      IF(IPAGE.EQ.1)  LGRP(4,IGRP) = LGRP(4,IGRP) + 1
      DO 250 INDEX = 1, 50
      ITASK = INDEX + JTPAGE
      IVAL = KTASK(ITASK)
      IF(IVAL.LE.0) GO TO 250
      ICORE(IVAL,INDEX,IGRP) = ICORE(IVAL,INDEX,IGRP) + 1
250   CONTINUE
300   CONTINUE
      WRITE(LOU) (((ICORE(I,J,K),I=1,NVALS),J=1,50),K=1,NGRPS)
400   CONTINUE
      ENDFILE LOU
      REWIND LOU
      REWIND LIN
      ISW = LIN
      LIN = LOU
      LOU = ISW
C ••••••••••••• ADVANCE RESIDENT CASE  •••••••••••••
207   CONTINUE
      CALL READF(INF,MBUF,$501)
      DECODE(804,MBUF) V001,LTASK,IDCARD
      IF(IDCASE.EQ.V001) GO TO 207
500   CONTINUE
501   CONTINUE
      CALL CLOSEF(INF)
      LDIST = (1-IFREQ)*NVALS+(1-IOTH)+3*(1-ISTAT)+1
      IF(IFREQ.EQ.1) LDIST = LDIST + 1
      NDPP = 54/LDIST
      IF(IDEBUG.EQ.'2') WRITE(6,844) LDIST
844   FORMAT(' LDIST = ',I6)
      NDIS  = NDPP - (NHEAD+5+LDIST)/LDIST
      NDIST = NDPP
      IF(IDEBUG.EQ.'2') WRITE(6,845) NDIS, NDIST
845   FORMAT(' NDIS = ',I6,'  NDIST = ',I6)
      LDUTY = IRA
      IDUTY = 1
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      MTASK = 0
      IPAGE = 0
      ITASK = 0
      IPTASK = 0
      NDEL = 0
      DO 539 I = 1, 26
      IF(LATASK(I).NE.IR ) NDEL = NDEL + 1
539   CONTINUE
C ******** BEGIN PAGING TASKS FOR PRINTING **********
502   CONTINUE
      READ(LIN) (((ICORE(I,J,K),I=1,NVALS),J=1,50),K=1,NGRPS)
      DO 599 JTASK = 1, 50
      ITASK = ITASK + 1
      IF(ITASK.GT.NTASK) GO TO 950
      MTASK = MTASK + 1
      IF(ITASK.LE.FLD(18,18,ITKDY(IDUTY))) GO TO 505
      IDUTY = IDUTY + 1
      LDUTY = LDUTY + 1
      MTASK = 1
505   CONTINUE
      IF(NDEL.EQ.0) GO TO 530
      DO 536 I = 1, NDEL
      IF(LDUTY.NE.LATASK(I)) GO TO 536
      GO TO 530
536   CONTINUE
      GO TO 599
530   IPTASK = IPTASK + 1
      IF(MOD(IPTASK,NDIS).NE.1) GO TO 540
      IPAGE = IPAGE + 1
      WRITE(6,806) ITL,IDREP,IPAGE
      IF(IREP.NE.1) WRITE(IRP,806) ITL,IDREP,IPAGE
806   FORMAT(1H1,8A6,28X,A6,' PAGE ',I3,/)
      IF(IPAGE.NE.1) GO TO 503
      NDIS = NDIST
      IF(NHEAD.NE.0) WRITE(6,807) ((IHEAD(I,J),I=1,14),J=1,NHEAD)
      IF(NHEAD.NE.0.AND.IREP.NE.1) WRITE(IRP,807) ((IHEAD(I,J),I=1,14)
     1 ,J = 1, NHEAD)
807   FORMAT(1X,13A6,A2)
      WRITE(6,808)
      IF(IREP.NE.1) WRITE(IRP,808)
808   FORMAT(1X)
      WRITE(6,809) (LGRP(1,J),J=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,809) (LGRP(1,J),J=1,NGRPS)
      WRITE(6,810) (LGRP(3,J),J=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,810) (LGRP(3,J),J=1,NGRPS)
      WRITE(6,811) (LGRP(4,J),J=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,811) (LGRP(4,J),J=1,NGRPS)
809   FORMAT(' GROUP IDENT.        ',12(3X,A6))
810   FORMAT(' GROUP MEMBERS       ',12I9)
811   FORMAT(' FACTOR SUBGROUP     ',12I9)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      GO TO 540
503   CONTINUE
      WRITE(6,809) (IDG(I),I=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,809) (IDG(I),I=1,NGRPS)
540   CONTINUE
      WRITE(6,808)
      IF(IREP.NE.1) WRITE(IRP,808)
      DO 504 IGRP = 1, NGRPS
      LGRP(1,IGRP) = 0
      LGRP(2,IGRP) = 0
      LGRP(3,IGRP) = 0
504   CONTINUE
      DO 510 IVAL = 1, NVALS
      IVT = IVAL + MIN - 1
      IVTSQ = IVT*IVT
      DO 509 IGRP = 1, NGRPS
      NRES = ICORE(IVAL,JTASK,IGRP)
      LGRP(1,IGRP) = LGRP(1,IGRP) + NRES*IVT
      LGRP(2,IGRP) = LGRP(2,IGRP) + NRES*IVTSQ
      LGRP(3,IGRP) = LGRP(3,IGRP) + NRES
509   CONTINUE
      IF(IFREQ.EQ.1) GO TO 510
      IF(IVAL.NE.1) GO TO 508
      WRITE(6,812)LDUTY,MTASK,IVT,(ICORE(IVAL,JTASK,IGRP),IGRP=1,NGRPS)
      IF(IREP.NE.1)WRITE(IRP,812)LDUTY,MTASK,IVT,(ICORE(IVAL,JTASK,IGRP)
     1 ,IGRP = 1, NGRPS)
812   FORMAT(1X,1H1,I3,7X,I2,6X,12I9)
      GO TO 510
508   CONTINUE
      WRITE(6,813) IVT,(ICORE(IVAL,JTASK,IGRP),IGRP=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,813) IVT,(ICORE(IVAL,JTASK,IGRP),
     1 IGRP = 1, NGRPS)
813   FORMAT(5X,7X,I2,6X,12I9)
510   CONTINUE
      IF(IFREQ.EQ.1) WRITE(6,812) LDUTY, MTASK
      IF(IFREQ.EQ.1.AND.IREP.NE.1) WRITE(IRP,812) LDUTY, MTASK
      IF(ISTAT.EQ.1) GO TO 520
      WRITE(6,833) (LGRP(3,IGRP),IGRP=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,833) (LGRP(3,IGRP),IGRP=1,NGRPS)
833   FORMAT(7X,'VALID N',6X,12I9)
520   CONTINUE
      IF(IOTH.EQ.1) GO TO 512
      DO 511 I = 1, NGRPS
      KOTH(I) = LGRP(4,I) - LGRP(3,I)
511   CONTINUE
      WRITE(6,821) (KOTH(I),I=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,821) (KOTH(I),I=1,NGRPS)
821   FORMAT(9X,'OTHER',6X,12I9)
512   CONTINUE
      IF(ISTAT.EQ.1) GO TO 521
```

LINE #   TEXT...

```
      DO 513 IGRP = 1, NGRPS
      X = LGRP(3,IGRP)
      SUM = LGRP(1,IGRP)
      SUMSQ = LGRP(2,IGRP)
      XMEAN = SUM/X
      SD = SQRT(ABS(SUMSQ/X - XMEAN*XMEAN))
      GPM(IGRP) = XMEAN
      GPS(IGRP) = SD
  513 CONTINUE
      WRITE(6,823) (GPM(I),I=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,823) (GPM(I),I=1,NGRPS)
  823 FORMAT(10X,'MEAN',6X,12F9.4)
      WRITE(6,824) (GPS(I),I=1,NGRPS)
      IF(IREP.NE.1) WRITE(IRP,824) (GPS(I),I=1,NGRPS)
  824 FORMAT(10X,'S.D.',6X,12F9.4)
  521 CONTINUE
  599 CONTINUE
      GO TO 502
  950 CONTINUE
      CALL RESET
      IF(IREP.NE.1) CALL REPEND(ID1,IDREP,ITL,JSEQ)
      GO TO 3
  900 CONTINUE
      CALL ERRMSG(8,'*** INVALID NUMERIC IN ABOVE CARD.  SET SKIPPED.')
  920 CONTINUE
      CALL ERRMSG(8,'*** FOLLOWING CARDS SKIPPED DUE TO ABOVE ERROR.')
  921 READ(5,830,END=999) KSKIP
  830 FORMAT(A6)
      CALL PRINTC
      IF(KSKIP.NE.'END') GO TO 921
      GO TO 3
  903 WRITE(6,840)
  840 FORMAT(' ENCOUNTERED END-OF-FILE ON J.D. FILE.  RUN ABORTS.')
      CALL ERTRAN(2)
  904 CONTINUE
      WRITE(6,831) JSEQ, ISEQ
  831 FORMAT(' ENCOUNTERED REPORT #',I4,' WHILE SEARCHING FOR',I5,
     1 '.  REPORT IS BYPASSED.')
      GO TO 3
  905 CONTINUE
      WRITE(6,843)
  843 FORMAT(' MASK ON J.D. FILE HAD MORE THAN 7000 CASES.  REPORT',
     1 ' IS BYPASSED.')
      GO TO 3
  906 CONTINUE
      WRITE(6,832) JDA(3), NCASE
  832 FORMAT(' MASK ON J.D. FILE MADE WITH ',I5,' CASES.  MOUNTED',
     1 ' FILE HAS ',I5,' CASES.  REPORT IS BYPASSED.')
      GO TO 3
  909 CONTINUE
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      WRITE(6,841) IDCASE
841   FORMAT(' STARTING CARD NOT FOUND FOR CASE ',A4,' RUN ABORTS.')
      CALL ERTRAN(2)
999   CONTINUE
      CALL ERRMSG(3,'*** END OF ASFACT')
      STOP
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      COMMON I, GRPA(850), GRPB(850), IHOGSW, ISAME, HDTYPE, IX, PAGE
      COMMON /DATUMS/ ESAVE(6), ITSAVE(6), EFLAGS(6)
      COMMON  /TEMPOS/ IDA(850), IDB(850), IDADSP(850), IDBDSP(850)
      COMMON  /TEMPOS/ KPHA(8,850), KPHB(8,850)
      COMMON /TEMPOS/ AAPTS(1726),BAPTS(1726),AMEMPF(1726),BMEMPF(1726)
      COMMON  /TEMPOS/ D(1726), NMASKA(290), ICMUNA(50), ICMUNB(50)
      COMMON  /TEMPOS/                IVAR(3), ISQUAR(3), CONIV(3)
      COMMON  /TEMPOS/ ITYPV(3), DTORG(3), CUTV(3), ASUMD(3), ASSUMC(3)
      COMMON /TEMPOS/ CONSTK(3)
      COMMON  /TEMPOS/ NMETSK(1726), XCUTV(3)
      DIMENSION  ITABA(1), ITABB(1)
      DIMENSION  ITITLE(1),  ESUMM(12, 850), JESUMM(12, 850), ITABLE(1)
      DIMENSION   JMASK(6), KMASK(6)
C
      EQUIVALENCE (AAPTS,ITITLE),(ITABLE,AMEMPF(911)),(ITABA,BMEMPF(136))
     1,(ITABB,BMEMPF(637)),(ESUMM,IDA),(ESUMM,JESUMM),(BLNKS,IBLNKS)
      EQUIVALENCE (AAID,ID)
      DATA IGROUP /' GROUP'/
      DATA  IW /6HW     /, A /6HA     /,  B /6HB     /, IM /6HM     /
      DATA  KD, KG, KW  /6HD     , 6HG     , 6HW     /
      DATA IK,MDTSK,MDDTY/'0','TASK','DUTY'/
      DATA  PRINTE /6HPRINT /
      DATA MYSEQ1, MYSEQ2/' SEQ',' PAIR'/
      DATA  PRGMID  /6HAUTOJT/, ASTERS /6H******/
      DATA BLNKS/6H      /, STARS/6H$$$$$$/, TASKS/6HTASKS /
      DATA DUTIES/6HDUTIES/, IEOF/6HEND OF/, IMASK/07777777777700/
      DATA  IDTMSK /0770000000000/
      DATA   JMASK    /07777777777777, 00077777777777, 00000777777777,
     1 00000000777777, 00000000007777, 00000000000077/, KMASK    /6H999999,
     2 6H099999, 6H009999, 6H000999, 6H000099, 6H000009/
C
C
C
      DATA  ASUMD   /1.0, 1.0, 1.0/
      DATA ASSUMC  /39.0625, 156.25, 10000.0/
C
      DATA   IONE, ITWO, ITHREE /1, 2, 3/
C
C
C
      INTEGER OPT1, OPT2, OPT3, PAGE
      INTEGER  CUTV, CUTOFF
      INTEGER CUTV2
C
      REAL     LARGE, LARGEB
C
C
C
C
C              INITIALIZE YOUR CONSTANTS
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          PAGE = 0
          IX = MYSEQ1
          IHDGSW = 1
          CALL SETUP(10,27,0,0,0,0,20,21,0,0,0,JSEQ)
          IP = 10
          IT = 27
          IS = 20
          IR = 21
C
          READ(5,1000) OPT1, OPT2, OPT3, IVAR(1),
         1 IVAR(2), IVAR(3), CONIV(1), ISQUAR(1), ITYPV(1), DTORG(1),
         2 CONSTK(1), CUTV(1), XCUTV(1), CONIV(2), PRT, ISQUAR(2), ITYPV(2),
         3 DTORG(2), CONSTK(2), CUTV(2), XCUTV(2), CONIV(3), ISQUAR(3),
         4 ITYPV(3), DTORG(3), CONSTK(3), CUTV(3), XCUTV(3), CONST2, CNSTK2,
         5 GEE2, CUTV2, CONST3, GEE3
C
C                 NPAIR = NUMBER OF PAIRS OF BIN JOB DESC REQUEST CARDS
C                 L = NCASE OF STUDY
C                 NREL = 'R' FOR MULTI-REEL INPUTS
C                 OPT1, OPT2, OPT3 = OPTIONS TO BE SELECTED
C                 IVAR = VARIANTS TO BE SELECTED FOR OPTION 1
C                 CONIV = C - CONSTANT FOR OPTION 1
C                 ISQUAR = 1 FOR ABS VALUE OF D, 0 OR ' ' FOR D SQUARED
C                 ITYPV = D, G, OR W FOR DENOMINATOR CONSTANT TYPE
C                 DTORG = DENOMINATOR CONSTANT VALUE
C                 CONSTK = K CONSTANT FOR G-TYPE DENOMINATORS
C                 CUTV = NR OF ITERATIONS TO RUN BEYOND PEAK E-VALUE
C                 XCUTV = VALUE BELOW WHICH D'S ARE NOT TO BE SOUGHT
C
C
 1000     FORMAT(14X, 6I1, F10.0, I1, A1, 2F10.0,I3, 2F10.0,
         1 A5, /, I1, A1, 2F10.0, I3, 2F10.0, I1, A1, 2F10.0, I3, F10.0, /,
         2 3F10.0, I3, 2F10.0)
C
C
  125     IF((OPT1 .EQ. 1).OR. (OPT2 .EQ. 2) .OR. (OPT3.EQ.3)) GO TO 130
          WRITE(6,1006)
 1006     FORMAT(/ ,16X,'NO LEGAL COMPUTATIONAL OPTIONS SPECIFIED.')
          GO TO 99999
C
C
C
  130     CONTINUE
          DO 140 I = 1, 850
          IDADSP(I) = 0
          IDBDSP(I) = 0
C
          READ(5,1007,END=1776) GRPA(I),(KPHA(J,I),J=1,8),IDA(I)
          NPAIR = I
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
         READ(5,1007,END=1777) GRPB(I),(KPHB(J,I),J=1,8),IDB(I)
  140    CONTINUE
  1007   FORMAT(4X,A6,8A6,T77,I4)
C
C
  1777   NPAIR = NPAIR - 1
C
  1778   CONTINUE
C
         REWIND IP
         READ (IP) ICM,(ICMUNA(I),I=1,ICM)
         NTASK = ICMUNA(12)
         NDUTY = ICMUNA(9)
         NFOREL = NTASK + NDUTY
C
  300    READ (IP) ISUB, JSUB
         IF (JSUB .NE. IEOF) GO TO 300
C
         READ (IP) NRWDS,(ITITLE(I),I=1,NRWDS)
C
         ISHF18 = 2 ** 18
         K = 1
C
C
C
C
  399    DO 400 J = 1, NFOREL
C
C
         ISUB = ITITLE (K) / ISHF18
         JSUB = MOD(ITITLE(K), ISHF18)
         DECODE(6666,ITITLE(JSUB+1)) NMETSK(J)
  6666   FORMAT(A5)
C
C           PICK UP TASK / DUTY TITLE AND ADD IN TRAILING BLANK
C
         K = ISUB + 1
         IF (K .LT. NRWDS)  GO TO 400
C
         READ (IP) NRWDS,(ITITLE(I),I=1,NRWDS)
         K = 1
C
  400    CONTINUE
C
         REWIND IP
         CALL FREE(IP)
C
C           TABLES ARE NOW LOADED AND INITIALIZED
C                START PASSING FILE TO DISK
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          J = 0
          ILAST = 0
          NRFND = 0
          REWIND IT
          REWIND IS
C
C
C
  150     READ(IT,END=90000) ID, NTD, (AAPTS(I),I=1,NTD), NC35,
         1(NMASKA(I),I=1,NC35), JSEQ, ICM, (ICMUNA(I),I=1,ICM),
         2(D(I),I=1,NTD), (AMEMPF(I),I=1,NTD)
  155     DO 160  I = 1, NPAIR
C
          IF ((JSEQ.EQ.IDA(I)) .OR.(JSEQ.EQ. IDB(I))) GO TO 170
          GO TO 160
C
  170     NRFND = NRFND + 1
C
          IF (JSEQ.EQ.IDB(I))GO TO 190
          IF (J .NE. JSEQ)  GO TO 180
          IDADSP (I) = ILAST
          IF(GRPA(I).NE.' ') GO TO 200
          GRPA(I) = AAID
          IF(KPHA(1,I).NE.' ') GO TO 200
          K = ICMUNA(35)
          IF(K.NE.'JOBGRP'.AND.K.NE.'JOBSPC'.AND.K.NE.'SAMSEL') GO TO 200
          ENCODE(161,KPHA(1,I)) (ICMUNA(K),K=37,44)
  161     FORMAT(8A6)
          GO TO 200
C
  180     ILAST = ILAST + 1
          IDADSP (I) = ILAST
          IF(GRPA(I).NE.' ') GO TO 200
          GRPA(I) = AAID
          IF(KPHA(1,I).NE.' ') GO TO 200
          K = ICMUNA(35)
          IF(K.NE.'JOBGRP'.AND.K.NE.'JOBSPC'.AND.K.NE.'SAMSEL') GO TO 200
          ENCODE(161,KPHA(1,I)) (ICMUNA(K),K=37,44)
          GO TO 200
C
  190     IF (J .NE. JSEQ)  GO TO 195
          IDBDSP (I) = ILAST
          IF(GRPB(I).NE.' ') GO TO 200
          GRPB(I) = AAID
          IF(KPHB(1,I).NE.' ') GO TO 200
          K = ICMUNA(35)
          IF(K.NE.'JOBGRP'.AND.K.NE.'JOBSPC'.AND.K.NE.'SAMSEL') GO TO 200
          ENCODE(161,KPHB(1,I)) (ICMUNA(K),K=37,44)
          GO TO 200
C
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
195   ILAST = ILAST + 1
      IDBDSP (I) = ILAST
      IF(GRPB(I).NE.' ') GO TO 200
      GRPB(I) = AAID
      IF(KPHB(1,I).NE.' ') GO TO 200
      K = ICMUNA(35)
      IF(K.NE.'JOBGRP'.AND.K.NE.'JOBSPC'.AND.K.NE.'SAMSEL') GO TO 200
      ENCODE(161,KPHB(1,I)) (ICMUNA(K),K=37,44)
200   IF (J .EQ. JSEQ) GO TO 160
C
C
      ISUB = NTASK
      ISAME = 0
C
C          FORM DUTY (CUMULATIVE) VALUES FOR  TAPES NOT HAVING THIS INFO
C
      DO 10027 K = 1, NTASK
      DECODE(6667,NMETSK(K)) JSUB
6667  FORMAT(A1)
      IF (ISAME .EQ. JSUB)  GO TO 10022
C
C
      ISAME = JSUB
      ISUB = ISUB + 1
      AAPTS (ISUB) = 0.0
C
10022 AAPTS (ISUB) = AAPTS (ISUB) + AAPTS (K)
10027 CONTINUE
10028 CONTINUE
      WRITE(IS) ID,JSEQ ,(AAPTS(L),L=1,NFOREL), (AMEMPF(L),L=1,
     1  NFOREL), (ICMUNA(L), L = 1, ICM)
C
      J = JSEQ
C
160   CONTINUE
      IF (NRFND .NE. (NPAIR * 2)) GO TO 150
C
C        NOW THE ENTIRE USABLE PORTION OF THE BINARY JOB DESCRIPTION
C             FILE IS ON DISK...
C        THE DISPLACEMENT OF EACH DESCRIPTION NAMED IN IDA(I) IS IN
C             IDADSP(I), LIKEWISE FOR IDB(I) AND IDBDSP(I)...
      REWIND IR
      REWIND IS
      REWIND IT
      CALL FREE(IT)
C
500   DO 10000  I = 1, NPAIR
C
C
      DO  10005 J = 1, 6
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
        ITSAVE (J) = 0
 10005 ESAVE (J)  = 0.0
C
C
C           RESET DISK FILE POSITION
C
        REWIND IS
        JSEQA = IDADSP(I)
        IF(JSEQA.EQ.1) GO TO 115
        ISKIP = JSEQA - 1
        DO 114 JSKIP = 1, ISKIP
 114    READ(IS)
 115    CONTINUE
C
        READ (IS)AID,JSEQA,(AAPTS(L),L=1,NFOREL), (AMEMPF(L),L=1,
       1  NFOREL), (ICMUNA(J), J = 1, ICM)
C
        IF (JSEQA .EQ. IDA(I))  GO TO 10010
        WRITE (6, 1025) A, JSEQA, A, IDA (I)
 1025   FORMAT (' TROUBLE AT THE OK CORAL, JSEQ', A1, ' = ', I6, ', ID',
       1 A1, ' = ', I6, '.')
        GO TO 99999
C
C
 10010 JSEQB = IDBDSP(I)
        REWIND IS
        IF(JSEQB.EQ.1) GO TO 117
        ISKIP = JSEQB - 1
        DO 116 JSKIP = 1, ISKIP
 116    READ(IS)
 117    CONTINUE
C
        READ (IS)BID,JSEQB,(BAPTS(L),L=1,NFOREL), (BMEMPF(L),L=1,
       1  NFOREL), (ICMUNB(J), J = 1, ICM)
C
        IF (JSEQB .EQ. IDB(I))  GO TO 10020
C
        WRITE (6, 1025)  B, JSEQB, B, IDB (I)
        GO TO 99999
C
 10020 CALL LINES(99)
        CALL LINES(04)
        WRITE(6,1012) IK
 1012   FORMAT (A1, 15X, 109('*'))
        WRITE (6, 1013) A, GRPA(I), (KPHA(J,I),J=1,8), ICMUNA(17)
        WRITE (6, 1013) B, GRPB(I), (KPHB(J,I),J=1,8), ICMUNB(17)
        WRITE (6, 1012) BLNKS
C
 1013    FORMAT(16X,'JOB DESCRIPTION*',A1,'* ',A6, 4X
       1,8A6,'  NUMBER OF MEMBERS = ',I6)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
C
      ANRMEM = ICMUNA (17)
      BNRMEM = ICMUNB (17)
      IPGSAV =  PAGE
C
C
C
10029 IF (OPT1 .NE. 1) GO TO 12000
C
C
C
C
C
C            OPTION 1  VARIANTS ...
C                1 =  PERCENT TIME SPENT    -  TASKS
C                2 =  PERCENT TIME SPENT    -  DUTIES
C                3 =  PERCENT MEMBERS PERFORMING - TASKS
C
      DO   11000  IVRSUB = 1, 3
C
      IF  ( IVAR (IVRSUB) .NE. IVRSUB)  GO TO 11000
      J = IVAR(IVRSUB)
C
      IF (J .EQ. 2) GO TO 10030
C
      HDTYPE = HDTSK
      TTYPE = TASKS
C
C      FOR  TASK  CALCULATIONS
C
      ISUB = 1
C
      JSUB =  NTASK
C
      GO TO 10045
C
10030 ISUB = NTASK + 1
C
C
      JSUB = NFOREL
C
      TTYPE = DUTIES
      HDTYPE = HDDTY
C
10045 IF(ISQUAR(J).NE.0) ISQUAR(J) = 1
      IF(I.EQ.1) ISQUAR(J) = IABS(1-ISQUAR(J))
C
      ASSIGN 10050 TO II
      IF (ISQUAR(J) .EQ. 1) ASSIGN 10070 TO II
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
      DO 10050  K = ISUB, JSUB
C
      IF (J .EQ. 3) GO TO 10055
      D(K) = ABS (AAPTS(K) - BAPTS(K))
      GO TO I1, (10050, 10070)
10055 D (K) = ABS ((AMEMPF (K) / ANRMEM) - (BMEMPF (K) / BNRMEM))
     1        * 100.0
      GO TO I1, (10050, 10070)
C
10070 D(K) = D(K) *  D (K)
C
10050 CONTINUE
      CALL LINES(04)
      IF (J .EQ. 3) GO TO 10080
C
      WRITE (6, 1031) J, TTYPE
1031  FORMAT (/  , 42X, 'OPTION 1   VARIANT ', I1,' PER CENT TIME SPENT
     1- ', A6, //)
C
      GO TO 10150
C
10080 WRITE (6, 1030) J, TTYPE
C
1030  FORMAT (/  , 37X, 'OPTION 1   VARIANT ', I1, ' PER CENT MEMBERS PE
     1RFORMING - ', A6, //)
C
C
C
10150 IF (ITYPV(J) .EQ. IBLNKS)  ITYPV(J) = KG
      IF (ITYPV(J) .NE.  KD)  GO  TO  10160
C
      DCONST = DTORG (J)
      IF  (DCONST .EQ. 0.0) DCONST = ASUMD (J)
      GO TO 10200
C
10160 IF  (ITYPV(J) .NE. KG)  GO TO  10170
C
      IF (DTORG (J) .EQ. 0.0) DTORG (J) = 1.00
      IF (CONSTK (J) .EQ. 0.0) CONSTK (J) = 1.00
C
C
      DCONST = CONSTK (J) + SQRT (DTORG (J) * (ANRMEM + BNRMEM) /
     1    (ANRMEM * BNRMEM))
      GO TO 10200
C
C
C
10170 IF  (ITYPV(J) .NE. KW) GO TO 10180
      IF (DTORG (J) .EQ. 0.0) GO TO 10180
```

LINE #   TEXT...

```
 C
        DCONST = DTORG (J) /( AMIN1(DTORG (J), ANRMEM, BNRMEM))
 C
        GO TO 10200
 C
 C
 10180 CALL LINES(03)
       WRITE(6,1004) ITYPV(J), J
 1004  FORMAT ('0', 15X, 'ILLEGAL CONSTANT TYPE (',A1, ') SPECIFIED FOR
      1OPTION 1 - VARIANT ', I2, ', VARIANT IS BYPASSED', 20('*'))
       GO TO 11000
 C
 10200 C = CONIV (J)
       IF ( C .EQ. 0.0)  C = ASSUMC (J)
 C
       CUTOFF = CUTV (J)
       CUTOFX = XCUTV(J)
 C
       IF(CUTOFF .NE. 0) GO TO 10215
 10210 CUTOFF =  5
 C
 C
 10215 ANS = 0.0
 C
       ISUBX= 0
       LARGE = - 1.0E9
       ICNT  = 0
       DENOM  =  C  *  DCONST
       ISAME = 0
       EPREVS = 0.0
 C
 C
 C
 10250 IF ((I .GT. 1 ) .AND. (PRT .NE. PRINTE)) GO TO 10360
       IF (ISQUAR (J) .EQ. 1)  GO TO 10300
       CALL LINES(03)
       WRITE (6, 1028)  ITYPV(J)
 1028  FORMAT (38X, 'THE ABSOLUTE VALUE OF D WAS SPECIFIED, CONSTANT TYPE
      1 = ', A1, //)
       GO TO 10310
 C
 10300 CALL LINES(03)
       WRITE(6,1029) ITYPV(J)
 1029  FORMAT (44X, 'D SQUARED WAS SPECIFIED, CONSTANT TYPE = ',A1//)
 C
 10310 IF(ITYPV(J).NE.KG) GO TO 10350
       CALL LINES(1)
       WRITE(6,1044) CONSTK(J), DTORG(J)
 1044  FORMAT (37X, 'K CONSTANT =', F15.6, ', G CONSTANT =', F15.8)
 C
```

LINE #   TEXT...

```
10350 CALL LINES(2)
      WRITE(6,1014) C, DCONST, DENOM, CUTOFF, J
1014  FORMAT ('0', 7X,'C = ',F15.6,', D = ',F15.6,', C * D = ',
     1  F15.6, ', CUTOFF = ', I4, ', FOR OPTION 1 - VARIANT ', I1)
C
10360 IHDGSW = 1
      CALL LINES(6)
      WRITE(6,1005) MDTYPE, IGROUP,A,IGROUP,B
C
1005  FORMAT (/,'0 IT NR',2X,'LRG ',A6,4X,A6,X,A1,10X,A6,X,A1,
     110X, 'D VALUE', 7X, 'CUM SUM DIFF', 8X, 'SQUARE ROOT N', 7X,
     2 'E VALUE', ///)
      IHDGSW = 2
C
      K = JSUB - ISUB + 1
10400 DO 10450  J = ISUB, JSUB
C
      IF (D(J) .EQ. 0.0) GO TO 10450
C
      IF(D(J).LE.LARGE) GO TO 10450
      LARGE = D (J)
C
      ITASK = J
C
10450 CONTINUE
C
      IF  (ITASK .EQ. ISUBX) GO TO  10470
C
C
C
      ICNT  =  ICNT  + 1
C
C
      ANS = ANS + LARGE
      XCNT2 =  ICNT
      XCNT2 = SQRT (XCNT2)
      E =  ( ANS / XCNT2)        / DENOM
C
      EFLAG  = BLNKS
      IF  (E .GE. 1.0) EFLAG = STARS
C
      IF (IVAR(IVRSUB).GT. 2) GO TO 10455
      DVALUE = (AAPTS (ITASK) - BAPTS (ITASK))
      CALL LINES(01)
      WRITE (6, 1015) ICNT, NMETSK(ITASK), AAPTS(ITASK), BAPTS(ITASK),
     1  DVALUE,  ANS,XCNT2, E, EFLAG
C
C
1015  FORMAT ( 2X, I3, 5X, A5,  X, 6(F15.6, 3X), A6)
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
            GO TO 10457
      C
10455 AVALUE = (AMEMPF(ITASK) / ANRMEM) * 100.0
      BVALUE = (BMEMPF(ITASK) / BNRMEM) * 100.0
      DVALUE = AVALUE - BVALUE
      C
            CALL LINES(01)
            WRITE (6, 1015) ICNT, NMETSK(ITASK), AVALUE, BVALUE, DVALUE,
           1   ANS, XCNT2, E, EFLAG
      C
      C
      C
      C
10457 IF (E  .GT. EPREVS) GO TO 10458
            ISAME = ISAME + 1
            IF (D(ITASK)        .LT. CUTOFX) GO TO  10500
            GO TO 10459
      C
10458 IF (E .LT. ESAVE(IVRSUB)) GO TO 10459
            ESAVE(IVRSUB) = E
            ITSAVE(IVRSUB) = ICNT
            IF (D (ITASK) .LT. CUTOFX) GO TO 10500
      C
      C
10459 EPREVS = E
            LARGE = - 1.0E9
            D (ITASK) = 0.0
            IF (ISAME .GE. CUTOFF) GO TO 10500
      C
            ISUBX= ITASK
10460 IF  (ICNT  .LT. K)  GO TO 10400
      C
10470 CALL LINES(02)
            WRITE(6,1027) ICNT
 1027 FORMAT ('0',15X,'ALL ELEMENTS ZERO AFTER ', I4, ' ITERATIONS  --
     1FURTHER ITERATIONS QUASHED.')
      C
      C
      C
10500 CALL LINES(03)
            WRITE(6,1026) ICNT, IONE, IVAR(IVRSUB)
 1026 FORMAT (/,'0',15X,15('*'),'CUTOFF SATISFIED AFTER ',I4,  ' ITERATI
     1ONS FOR OPTION ', I1, ' - VARIANT ',I1, 15('*'))
      C
            IHDGSW = 1
            WRITE (6, 1017)
      C
      C                          SKIP SOME LINES AND PROCEED
 1017 FORMAT (///)
            CALL LINES(03)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C
11000 CONTINUE
C
C
C
12000 IF (OPT2 .NE. 2) GO TO 14000
C
C
      TTYPE = TASKS
      I1 = 1
      I2 = NTASK
      ISUBX = NTASK
C
12025 CALL LINES(05)
      WRITE(6,1033) TTYPE
1033  FORMAT ('0', 39X, 'OPTION 2, CUMULATIVE PER CENT TIME SPENT ON ',
     1 A6, ///)
C
      IF (TTYPE .EQ. TASKS) GO TO 12030
      IF (CONST2 .EQ. 12.50) CONST2 = 18.75
      GO TO 12035
12030 IF (CONST2 .EQ. 0.0) CONST2 = 12.50
12035 IF (CNSTK2 .EQ. 0.0) CNSTK2 = 1.00
      IF (GEE2 .EQ. 0.0) GEE2 = 1.00
      C = (CNSTK2 + SQRT (GEE2 * (ANRMEM + BNRMEM) /
     1                           (ANRMEM * BNRMEM))) * CONST2
C
      IF (CUTV2 .EQ. 0) CUTV2 = 5
C
      ISAME= 0
      JSAME = 0
      EPREVS = 0.0
      ICNT = 0
C
      TSUMA = 0.0
      TSUMB = 0.0
      J = 0
C
C                      OUTER LOOP FOR ALL TASKS
C
C
      LARGE = 0.0
      LARGEB = 0.0
      IF ((I .GT. 1) .AND. (PRT .NE. PRINTE)) GO TO 12040
      CALL LINES(03)
      WRITE (6, 1023) CONST2, CNSTK2, GEE2, C
C
1023  FORMAT (15X, 'C = ', F15.6, ', K = ', F15.6, ', G = ', F15.6,
     1 ', DENOMINATOR = ', F15.6, ' FOR OPTION 2.'//)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
12040 IHOGSW = 1
      CALL LINES(06)
      WRITE (6, 1032) A, B, A, B
1032  FORMAT (/,'0',2X,'ITERATION',3X,'LARGEST ',A1,3X, 'LARGEST ',
     1 A1, 3X, 'CUMULATIVE SUM ', A1, 5X, 'CUMULATIVE SUM ', A1, 8X,
     2 'ABSOLUTE DIFF', 10X, 'E  VALUE' ///)
C
      IHOGSW = 3
12050 J = J + 1
      IF(LARGE .EQ.0.0) LARGE  = -1.0
      IF(LARGEB.EQ.0.0) LARGEB = -1.0
C
      ISUB = 0
      JSUB = 0
      DO 12200  K = I1, I2
C
      IF(AAPTS(K).LE.LARGE ) GO TO 12100
      LARGE = AAPTS(K)
      ISUB = K
C
12100 IF(BAPTS(K).LE.LARGEB) GO TO 12200
      LARGEB = BAPTS (K)
      JSUB = K
12200 CONTINUE
C
      IF ((ISAME .EQ. ISUB) .AND.(JSAME .EQ. JSUB)) GO TO 13400
C
      AAPTS (ISUB) = -5.0
      BAPTS (JSUB) = -5.0
C
      TSUMA = TSUMA + LARGE
      TSUMB = TSUMB + LARGEB
      ABSTSM = ABS( TSUMA - TSUMB)
      E = ABSTSM / C
C
      EFLAG = BLNKS
      IF (E .GE. 1.0) EFLAG = STARS
      CALL LINES(01)
      WRITE (6, 1018) J, NMETSK (ISUB), NMETSK (JSUB), TSUMA, TSUMB,
     1  ABSTSM, E, EFLAG
C
1018  FORMAT (6X,I3,8X,A5, 7X,A5, 2X, F15.6, 5X, F15.6, 9X, F15.6, 5X,
     1   F15.6, 6X, A6)
C
C
      LARGE = 0.0
      LARGEB = 0.0
C
      IF (E .GT. EPREVS) GO TO 12300
```

LINE # TEXT...

```
       ICNT = ICNT + 1
       IF (ICNT .GE. CUTV2) GO TO 13500
       GO TO 12400
C
12300 IF (TTYPE .EQ. DUTIES) GO TO 12350
       IF (ESAVE (4) .GT. E) GO TO 12400
       ESAVE(4) = E
       ITSAVE(4) = J
       GO TO 12400
12350 IF (ESAVE (5) .GT. E) GO TO 12400
       ESAVE (5) = E
       ITSAVE (5) = J
C
C
12400 EPREVS = E
C
       JSAME = JSUB
       ISAME = ISUB
       IF (J .LT. ISUBX) GO TO 12050
       GO TO 13500
C
C
13400 WRITE (6,1027)  J
C
13500 IHDGSW = 1
       CALL LINES(06)
       WRITE (6, 1016)  J, ITWO
C
1016  FORMAT (/,'0',16X,15('*'),'CUTOFF SATISFIED AFTER ',I4,  ' ITERATI
     1ONS FOR OPTION ', I1, 15('*')////)
C
       IF (TTYPE .EQ. DUTIES) GO TO 13600
C
       I1 = NTASK + 1
       I2 = NFOREL
       TTYPE = DUTIES
       ISUBX = NDUTY
       GO TO 12025
C
13600 WRITE (6, 1017)
       CALL LINES(3)
C
C              SKIP SOME LINES AND CHECK THE LAST OPTION
C
14000 IF (OPT3 .NE. 3) GO TO 14700
C
       CALL LINES(5)
       WRITE (6, 1034)
1034  FORMAT ('0', 41X, 'OPTION 3  AVERAGE NUMBER OF TASKS PERFORMED',
     1 ////)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
C
14100 TSUMA =0.0
      TSUMB = 0.0
      IF (CONST3 .EQ. 0.0) CONST3 = 1.50
      IF (GEE3 .EQ. 0.0) GEE3 = 1.00
      C = CONST3 + SQRT ((GEE3 * (ANRMEM + BNRMEM)) /  (ANRMEM *
     1  BNRMEM))
      DO  14200 J = 1, NTASK
      TSUMA = TSUMA + AMEMPF (J)
14200 TSUMB =  TSUMB  + BMEMPF(J)
C
      TSUMA = TSUMA / ANRMEM
      TSUMB = TSUMB / BNRMEM
C
      IF (TSUMB .GT. TSUMA) GO TO 14400
      LARGE = A
      LARGEB = TSUMA / TSUMB
C
      GO TO 14500
C
14400 LARGE = B
      LARGEB = TSUMB / TSUMA
C
14500 E =  LARGEB / C
C
C
C
      EFLAG = BLNKS
      ESAVE (6) = E
      IF (E .GT. 1.0) EFLAG = STARS
      IF ((I .GT. 1 ) .AND. (PRT .NE. PRINTE)) GO TO 14595
      CALL LINES(1)
14590 WRITE (6, 1021) CONST3, GEE3, C
1021  FORMAT (10X, 'OPTION 3 CONSTANTS USED --  K = ', F15.6, ', G = ',
     1 F15.6, ', C = ', F15.6, '.')
C
14595 IF (LARGE .EQ. B) GO TO 14600
      CALL LINES(6)
      WRITE (6, 1019) LARGE, TSUMA, TSUMB, LARGEB,            E, EFLAG
C
1019  FORMAT (//,10X,'LARGEST AVE. NR TASKS PERFORMED WAS JOB DESCRIPTIO
     1N ',A1,',',8X, 'AVE OF LARGEST = ', F15.6, ///,10X,        'AVE OF
     2 SMALLEST = ', F15.6, ',', 5X, 'MLARGE  / MSMALL = ', F15.6, ',',
     3 10X, 'E VALUE =', F15.6, 2X, A6)
C
      GO TO 14700
C
14600 CALL LINES(6)
      WRITE(6,1019) LARGE, TSUMB, TSUMA, LARGEB, E, EFLAG
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
C          PUT OUT RECORD OF E VALUES, PEAK AND THE IT NR ON WHICH
C                THEY OCCURRED
C
 14700 WRITE(IR) ( ITSAVE(J),ESAVE(J),J=1,5), ESAVE(6), IPGSAV
C
 10000 CONTINUE
C
C
C
C
       REWIND IR
       IHDGSW = 4
       HDTYPE = A
C
C
C
 18000 DO 18100 I = 1, NPAIR
 18100 READ(IR) (ESUMM(ISUB,I),ISUB=1,12)
C
C
C
C
C          TRANSFORM  GRPA+B (6 CHAR. REPORT ID'S)  INTO  NUMERIC-PORTION
C                ONLY TABLE FOR SORTS (ITABA,B)...
C          USED FOR ORDER OF PRINT-OUT IN E  VALUE SUMMARY
C
C
C
       DO  19000  I = 1, NPAIR
C
C
       DECODE(6668,GRPA(I)) ISUB
       CALL BCDBIN(ISUB,ISUB,4,$19100)
       GO TO 19150
 19100 DECODE(6669,GRPA(I)) ISUB
       CALL BCDBIN(ISUB,ISUB,3,$19101)
       GO TO 19150
 6668  FORMAT(2X,A4)
 6669  FORMAT(3X,A3)
 19101 ISUB = 0
C
 19150 ITABA(I) = ISUB
C
       DECODE(6668,GRPB(I)) ISUB
       CALL BCDBIN(ISUB,ISUB,4,$19200)
       GO TO 19250
 19200 DECODE(6669,GRPB(I)) ISUB
       CALL BCDBIN(ISUB,ISUB,3,$19201)
       GO TO 19250
```

LINE # TEXT...

```
19201 ISUB = 0
C
19250 ITABB(I) = ISUB
C
C
C
19000 CONTINUE
C
C
C
C                DONE WITH NUMERIC-ONLY TABLE GENERATION
C
      ISAME = 0
C
C
19290 DO 19300 I = 1, NPAIR
C
19300 ITABLE (I) = I
C
      ISUB = 0
C
19310 ISUB = ISUB + 1
      JSUB = ISUB + 1
C
      IF (JSUB .GT. NPAIR) GO TO 19400
C                                  NOW  SORT
19320 I1 = ITABLE (ISUB)
      I2 = ITABLE (JSUB)
C                                          BUBBLE
      IF (ISAME .EQ. 1)  GO TO 19330
C
C
      IF (ITABA (I1) - ITABA (I2)) 19310, 19310, 19340
C
C                  BUBBLE
C
19330 IF (ITABB (I1) - ITABB (I2)) 19310, 19310, 19340
C
C                                              BUBBLE
19340 JSAME = I2
      ITABLE (JSUB) = ITABLE (ISUB)
19350 ISUBX = ISUB - 1
C
19360 IF (ISUBX .GT. 0) GO TO 19365
      ITABLE (I) = JSAME
      GO TO 19310
C
19365 I = ITABLE (ISUBX)
C                                                   BUBBLE
      IF (ISAME .EQ. 1) GO TO 19370
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C                                                   GRP
      IF (ITABA (I) - ITABA (I2))      19380, 19380, 19390
19370 IF (ITABB (I) - ITABB (I2))      19380, 19380, 19390
C
19380 ITABLE (ISUBX + 1) = JSAME
      GO TO 19310
C                             BUBBLE BUBBLE  BUBBLE
19390 ITABLE (ISUBX + 1) = I
      ISUBX = ISUBX - 1
      GO TO 19360
C
C
19400 CONTINUE
C
      CALL LINES(99)
C
C
19500 DO  20200  ISUB = 1, NPAIR
C
      I1 = 0
      I = ITABLE (ISUB)
C
      DO  20110  J = 2, 12, 2
20100 I1 = I1 + 1
C
      JSUB = J
      IF (J .EQ. 12) JSUB = 11
      IF(ESUMM (JSUB ,I).GT. 1.0) GO  TO  20105
      EFLAGS(I1)= BLNKS
      GO TO 20110
C
20105 EFLAGS(I1)= STARS
C
20110 CONTINUE
C
C
C                       REPORT SUMMARY SORTED ON  GRPA
C
C
21150 WRITE (6, 1038) ISUB, JESUMM (12, I), GRPA (I), GRPB (I),
     1 JESUMM (1, I), ESUMM (2, I), EFLAGS (1), JESUMM (3, I),
     2 ESUMM (4, I), EFLAGS (2), JESUMM (5, I), ESUMM (6, I), EFLAGS(3),
     3 JESUMM (7, I), ESUMM (8, I), EFLAGS (4), JESUMM (9, I),
     4 ESUMM (10, I), EFLAGS (5), ESUMM (11, I), EFLAGS (6)
C
C
1038  FORMAT (X, I3, X, I4, 3X, A6, X, A6, 2X, 5(I4, F8.3, 2X, A4),
     1  2X, F8.3, 1X, A4)
C
C
```

LINE #   TEXT...

```
C
C
20200 CONTINUE
C
C
C          THE 3 SORT SEQUENCES AND  CORRESPONDING VALUES OF ISAME ARE...
C                 1    6-CHAR REPORT ID - GROUP A, ASCENDING ON NUMERICS
C                 2    6-CHAR REPORT ID - GROUP B, ASCENDING ON NUMERICS
C                 3    ORIGINAL CONTROL CARD ORDER
C
C
C
C
C                    NOW  FOR THE 'B' SEQUENCE
C
C
      ISAME = ISAME + 1
      MDTYPE = B
C
      IF (ISAME .EQ. 1) GO TO  19290
      IF (ISAME .EQ. 3)  GO  TO  30000
C
      DO  25000 I = 1, NPAIR
25000 ITABLE (I) = 1
C
      GO TO 19400
C
C
C
C
C     .  .  .  .     E      O    J       .  .  .  .
C
C
C
C
C
C
C
C
C
90000 IF(NRFND.EQ.(NPAIR*2)) GO TO 99999
      J = (NPAIR * 2) - NRFND
      WRITE (6, 1022)  J
1022  FORMAT(//,11X,  'SPECIFICATION ERROR - ', I4,   ' JSEQUENCE NUMBER
     1S NOT FOUND ON THIS INPUT TAPE.')
30000 CONTINUE
99999 WRITE (6, 1011)
C
1011  FORMAT('1END OF AUTOJT')
      STOP
      SUBROUTINE LINES(MORE)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      COMMON I,GPA(850),GPB(850),ISW,ISAME,MDT,IX,PAGE
      INTEGER PAGE
      DATA MYSEQ2 /' PAIR'/,LINE,PAGE / 0, 0 /
      DATA IG /' GROUP'/
 1    LINE = LINE + MORE
      IF(LINE.LE.56) RETURN
      PAGE = PAGE + 1
      LINE = 0
      IF(ISW.EQ.4) GO TO 50
      LINE = 3
      WRITE (6,1010) I, GPA(I), GPB(I), PAGE
 1010 FORMAT('1',9X,'AUTOMATED JOB - TYPE SELECTION  PROGRAM', 8X,'PAIR
     1NR ',I4,5X,'GROUPA = ',A6,5X,'GROUPB = ',A6,14X,'PAGE ',I4,//)
      GO TO (30, 10, 20), ISW
 10   WRITE(6,1005) MDT, IG, IG
 1005 FORMAT(//,'  IT NR',2X,'LRG ',A6,4X,A6,1X,'A',10X,A6,1X,'B',10X,
     1'D VALUE',7X,'CUM SUM DIFF',8X,'SQUARE ROOT N',7X,'E VALUE',///)
      LINE = LINE + 6
      GO TO 30
 20   WRITE(6,1032)
 1032 FORMAT(//,3X,'ITERATION',3X,'LARGEST A   LARGEST B   CUMULATIVE SU
     1M A      CUMULATIVE SUM B',8X,'ABSOLUTE DIFF',10X,'E  VALUE',///)
      LINE = LINE + 6
      GO TO 30
 50   WRITE(6,1035) PAGE
 1035 FORMAT('1',32X,'AUTOMATED JOB - TYPE SELECTION PROGRAM   E VALUE S
     1UMMARY',31X,'PAGE ',I4)
      LINE = 1
      IF(ISAME.EQ.2) GO TO 60
      WRITE(6,1042) MDT
 1042 FORMAT('0',48X,'SORTED ON GROUP ',A1,5H'S ID,//)
      LINE = LINE + 4
      GO TO 70
 60   WRITE(6,1043)
 1043 FORMAT('0',48X,'LISTED IN ORIGINAL CONTROL CARD ORDER',//)
      LINE = LINE + 4
      IX = MYSEQ2
 70   WRITE(6,1041) IX
 1041 FORMAT(27X,'PER CENT TIME',5X,'PER CENT TIME',5X,'PER CENT MEMBERS
     1  CUM PER CENT',6X,'CUM PER CENT',9X,'AVG NR OF',/,'0',30X,'TASKS'
     2,12X,'DUTIES',13X,'TASKS',12X,'TIME',14X,'TIME',14X,'TASKS PERF.',
     3/,A5,' PAGE',2(2X,'GROUP'),3X,'OPT 1 - VAR 1',5X,'OPT 1 - VAR 2',
     45X,'OPT 1 - VAR 3',5X,'OPT 2',3X,'TASKS',5X,'OPT 2  DUTIES',9X,
     5'OPT 3',/,2X,2('NR',3X),' A',6X,'B',6X,5('ITER  E VALUE',5X), 4X,
     6'E VALUE',///)
      LINE = LINE + 7
 30   IF(MORE.GT.30) RETURN
      LINE = LINE + MORE
      RETURN
      END
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      COMMON ISID1,ISID2,KE(500),KD(1726)
      DIMENSION AY(11500)
      NAVAIL=11500
      IMT=10
      CALL SETUP(IMT,26,0,0,0,0,20,0,0,0,0,JSEQ)
C
C     READ I.D. CARD
      READ(5,1) IPID,ISID1,ISID2
    1 FORMAT(A6,1X,A6,A2)
C
C     READ COMMUNICATION REGION
      READ(IMT) K,(KE(I),I=1,K)
      NTASK=KE(12)
      IMAX=NAVAIL/(NTASK*3+77)
C
C     SET UP STORAGE FOR VARIABLE DIMENSIONS
      N1 = 9*IMAX+1
      N2 = N1+36*IMAX
      N3 = N2+8*IMAX
      N4 = N3+9*IMAX
      N5 = N4+IMAX
      N6 = N5+NTASK*IMAX
      N7 = N6+NTASK*IMAX
      N8 = N7+NTASK*IMAX
      N9 = N8+IMAX
      N10 = N9+IMAX
      N11 = N10+IMAX
      N12 =N11+IMAX
      N13 = N12+IMAX
      N14 = N13+IMAX
      N15=N6
      N16=N5
      N17=N14+IMAX
      N18=N17+IMAX
      N19=N18+IMAX
      LGM=N19+IMAX
      CALL AVALL(IMAX,NTASK,AY,AY(N1),AY(N2),AY(N3),AY(N4),AY(N5),
     1 AY(N6),AY(N7),AY(N8),AY(N9),AY(N10),AY(N11),AY(N12),AY(N13),
     2 AY(N14),AY(N15),AY(N16),AY(N17),AY(N18),AY(N19),LGM,JSEQ)
      CALL MARGIN
      STOP
      SUBROUTINE AVALL(IMAX,NTASK,IVL,IVAR,IOPN,ISGT,NNN,SUMX,SUMX2,
     1 ACNT,MVAR,LVAR,ISIZE,ISKL,IBCD,IDSP,IBAD,SD,MEAN,NMEM,
     2 ALO,AHI,LGM,JSEQ)
      COMMON ISID1,ISID2,KE(500),KD(1726)
      DIMENSION IVL(9,IMAX),IVAR(36,IMAX),IOPN(8,IMAX),ISGT(9,IMAX),
     1 NNN(IMAX),SUMX(NTASK,IMAX),SUMX2(NTASK,IMAX),ACNT(NTASK,IMAX),
     2 MVAR(IMAX),LVAR(IMAX),ISIZE(IMAX),ISKL(IMAX),IBCD(IMAX),
     3 IDSP(IMAX),IBAD(IMAX),SD(NTASK,IMAX),MEAN(NTASK,IMAX),NMEM(IMAX),
     4 ALO(IMAX),AHI(IMAX)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      DIMENSION IREL(928),HEAD(14,6),IBUF(2,2),KTST(9),
     1 INUM(5),        KB(1726),ILAB(8),FBUF(2,2),ITSK(928),IHVBUF(928),
     2 INX(1726),IOUT(2),LBUF(54),JHI(3)
      EQUIVALENCE(IBUF,FBUF),(IV,VALU),(IREL,ITSK),(IHVBUF,KB),(KD,INX)
      DATA IPROG/6HAVALUE/,ICEE/6HC      /,IEND/6HEND    /,KEND/6HEND OF/
      DATA JHI/18H9      99     999   /
      REAL MEAN
      IHT=10
      ISCH=20
      CALL DISK(250,465)
      INUM(1)=2**30
      INUM(2)=2**24
      INUM(3)=2**18
      INUM(4)=2**12
      INUM(5)=2**6
      IOUT(1)=6
      IOUT(2)=26
      ITASK=NTASK-1
      IF(ISID1.EQ.KE(1).AND.ISID2.EQ.KE(2)) GO TO 105
      WRITE(6,2)
    2 FORMAT(34HIS.I.D. NO MATCH.  RUN TERMINATED.)
      GO TO 407
  105 NDICT=KE(4)
C
C     READ RELATIVE LOCATION TABLE.
  107 READ(IHT) K,(IREL(I),I=1,K)
      REWIND ISCH
C
C     SKIP DICTIONARY.
      DO 110 I=1,NDICT
  110 READ(IHT) K
C
C     READ AND CHECK CONTROL CARD.
      NJD=1
  112 READ(5,3,END=113) JEND,MVAR(NJD),LVAR(NJD),ISIZE(NJD),ISKL(NJD),
     1 ISEQ,IHEAD,IWT,IBCD(NJD),IDSP(NJD),ALO(NJD),AHI(NJD)
    3 FORMAT(A3,A1,I3,6I2,1X,A6,1X,F5.0,1X,F5.0)
      IF(ISKL(NJD).EQ.1) ISIZE(NJD)=1
      IF(IBCD(NJD).EQ.0) IBCD(NJD) = 2
      IF(ISEQ.GE.1.AND.ISEQ.LE.3) GO TO 115
      WRITE(6,15)IDSP(NJD)
   15 FORMAT(29H1INCORRECT SORT SEQUENCE FOR ,A6,13H.  ASSUMED 1.)
      ISEQ=1
      GO TO 115
  113 LEND=IEND
      IF(NJD.GT.1) GO TO 141
      GO TO 407
  115 IF(IHEAD.GE.1.AND.IHEAD.LE.6) GO TO 120
      WRITE(6,17)IDSP(NJD)
   17 FORMAT(41H1THE NUMBER OF TITLE CARDS SPECIFIED FOR ,A6,27H IN ERR
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
       10R.  RUN TERMINATED.)
        GO TO 407
  120   IF(IWT.LE.1) GO TO 121
        WRITE(6,19)IDSP(NJD)
   19   FORMAT(41H1WEIGHT OPTION INCORRECTLY SPECIFIED FOR ,A6,13H.  ASSU
       1MED 0.)
        IWT=0
  121   IF(IBCD(NJD).GE.1.AND.IBCD(NJD).LE.2) GO TO 125
        WRITE(6,21)IDSP(NJD)
   21   FORMAT(43H1BCD TAPE OPTION INCORRECTLY SPECIFIED FOR ,A6,13H.  AS
       1SUMED 1.)
        IBCD(NJD)=1
C
C       READ INDEX TITLE
  125   READ(5,6)(ILAB(I),I=1,8)
    6   FORMAT(8A6)
C
C       READ TITLE CARDS.
        READ(5,5)((HEAD(I,K),I=1,14),K=1,IHEAD)
    5   FORMAT(13A6,A2)
C
C       READ VARIABLE NAMES AND OPERATORS
        JERR=0
           IPRNT = 0
           CALL SAMPLE(NJD,ISGT(1,NJD),NN,JERR,IPRNT)
           NNN(NJD) = NN
        N=1
        DO 146 K=1,NN
        IF(ISGT(K,NJD).GE.4) GO TO 135
        IF(ISGT(K,NJD).GE.0) GO TO 130
C
C       READ LIMITS - ALPHANUMERIC
        READ 9,L,KTST(K),((IBUF(J,M),M=1,L),J=1,2)
    9   FORMAT(I2,2X,I3,1X,4A6)
        KTST(K)=-KTST(K)
        GO TO 140
C
C       READ LIMITS - INTEGER
  130   READ 11,L,KTST(K),IBUF(1,1),IBUF(2,1)
   11   FORMAT(I2,2X,I3,1X,I6,6X,I6)
        GO TO 140
C
C       READ LIMITS - FLOATING POINT
  135   READ 13,L,KTST(K),FBUF(1,1),FBUF(2,1)
   13   FORMAT(I2,2X,I3,1X,2F12.6)
  140   KL =L*2
        IF(ISGT(K,NJD).NE.KTST(K))JERR=JERR+1
C
C       TEST TO SEE IF ANY ERRORS IN VARIABLE ACTION CARDS
        IF(JERR.EQ.0) GO TO 142
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      WRITE(6,23)IDSP(NJD)
   23 FORMAT(54HIERROR IN VARIABLE INTERACTION OR DEFINITION CARD FOR ,
     1A6,15H REPORT SKIPPED)
      GO TO 112
  142 DO 145 M=1,L
      J=N+M-1
      IVAR(J,NJD)=IBUF(1,M)
      J=J+L
  145 IVAR(J,NJD)=IBUF(2,M)
      N=N+KL
      IVL(K,NJD)=L
  146 CONTINUE
C
C     WRITE CONTROL,INDEX TITLE, AND TITLE CARDS
      WRITE(ISCR)JEND,MVAR(NJD),LVAR(NJD),ISIZE(NJD),ISKL(NJD)
     1,ISEQ,IHEAD,IWT,IBCD(NJD),IDSP(NJD),ALO(NJD),AHI(NJD)
      WRITE(ISCR)(ILAD(I),I=1,8)
      WRITE(ISCR)((HEAD(I,K),I=1,14),K=1,IHEAD)
C
C     TEST IF MAX NUMBER OF GROUPS HAS BEEN REACHED.
      IF(NJD.GE.IMAX) GO TO 157
      NJD=NJD+1
      GO TO 112
C
  141 NJD=NJD-1
  157 REWIND ISCR
      DO 151 J=1,NJD
      DO 150 I=1,NTASK
      SUMX(I,J)=0.
      SUMX2(I,J)=0.
      ACNT(I,J)=0.
  150 CONTINUE
  151 CONTINUE
C
C     FIND REL LOC FOR BACKGROUND VARIABLES
      DO 152 I=1,NJD
      IF(MVAR(I).EQ.ICEE) GO TO 152
      IX=LVAR(I)
     LVAR(I)=MOD(IREL(IX),INUM(3))
  152 CONTINUE
      DO 153 I=1,NJD
      NMEM(I)=0
  153 IBAD(I)=0
C
C     READ CASE DATA AND MOVE.
  154 READ(IMT) NW1,(KE(I+1),I=1,NW1),NW2,(KD(I),I=1,NW2),NW3,
     1 (KB(I),I=1,NW3)
      IF(KE(2).EQ.KEND) GO TO 210
      KE(1)=NW1
      KE(NW1+2)=NW2
```

LINE #   TEXT...

```
      DO 155 I=1,NW2
      J=NW1+2+I
  155 KE(J)=KD(I)
C
C     TEST VARIABLES FOR THIS CASE
      DO 205 IJD=1,NJD
      IF(IBAD(IJD).NE.0) GO TO 205
      NN=NNN(IJD)
      DO 160 I=1,36
  160 LBUF(I)=IVAR(I,IJD)
      N=0
C
C     LOOP FOR VARIABLES
      DO 185 L=1,NN
      KL=ISGT(L,IJD)
C
C     NEG. KL, INPUT VARIABLE       POS. KL, COMPUTED VARIABLE
      IF(KL.GT.0) GO TO 180
      KL=-KL
C
C     TEST VARIABLE LENGTH
      KM=IREL(KL)/INUM(3)
      IF(KM.EQ.IVL(L,IJD)) GO TO 170
      WRITE(6,27)IDSP(IJD)
   27 FORMAT(31H1SPECIFIED VARIABLE LENGTH FOR ,A6,27H IN ERROR.  REPOR
     IT SKIPPED.)
      IBAD(IJD)=1
      GO TO 205
C
C     TEST IF INPUT VARIABLE IS IN RANGE
  170 KN=MOD(IREL(KL),INUM(3))+KM+1+KE(1)
      KS=KN-KM+1
      KLO=N+1
      KHI=KLO+KM
      KTST(L)=LASCMP(KE(KS),LBUF(KLO),LBUF(KHI),KM)
      N=N+KM+KM
      GO TO 185
C
C     TEST IF COMPUTED VARIABLE IS IN RANGE
  180 KTST(L)=0
      IF(LBUF(N+1).LE.KE(KL+1).AND.LBUF(N+2).GE.KE(KL+1))KTST(L)=1
      N=N+2
  185 CONTINUE
C
C     PERFORM LOGICAL COMBINATION OF VARIABLES
      JERR=0
     ITRA = 1
     CALL SELECT(IJD,KTST,$205,$187)
     ITRA = 2
     GO TO 190
```

```
LINE #   TEXT...

 187   CONTINUE
       WRITE(6,29)IDSP(IJD)
    29 FORMAT(36H1THE AND/OR OPERATORS INCORRECT FOR ,A6,18H.  REPORT SK
      1IPPED.)
       IBAD(IJD)=1
       GO TO 205
C
C      ITRA = 2 MEANS THIS CASE SELECTED
C
C      FIND BACKGROUND OR COMPUTED VARIABLE AND SUM INTO ARRAYS
   190 IX=LVAR(IJD)+1
       IF(MVAR(IJD).NE.ICEE) GO TO 192
       GO TO 207
   192 INN=ISIZE(IJD)
       IX=LVAR(IJD)+KE(1)+2
       IF(ISKL(IJD).EQ.0) GO TO 193
       DECODE(188,KE(IX)) ITEMP
   188 FORMAT(3X,A1)
       CALL BCDBIN(ITEMP,ITEMP,1,$205)
       VALU=ITEMP
       IF(VALU.EQ.1..OR.VALU.EQ.3..OR.VALU.EQ.5..OR.VALU.EQ.7..OR.VALU
     1 .EQ.9.) GO TO 194
       GO TO 205
   193 CALL BCDBIN(KE(IX),JVALU,INN,$205)
       VALU=JVALU
   207 IF(VALU.LT.ALO(IJD).OR.VALU.GT.AHI(IJD)) GO TO 205
   194 DO 195 J=1,NW3
       I=KB(J)/INUM(3)
       SUMX(I,IJD)=SUMX(I,IJD)+VALU
       SUMX2(I,IJD)=SUMX2(I,IJD)+VALU**2
   195 ACNT(I,IJD)=ACNT(I,IJD)+1.
       NMEM(IJD)=NMEM(IJD)+1
   205 CONTINUE
       GO TO 154
C
C      READ AND SAVE TASK TITLES AND LOCATOR TABLE.
   210 READ(IHT) K,(ITSK(I),I=1,K)
       IF(ITSK(1).NE.KEND) GO TO 210
       BACKSPACE IHT
       BACKSPACE IHT
       READ(IHT) K,(ITSK(I),I=1,K)
       IBST=ITSK(K)/INUM(3)+1
       DO 220 IXX=1,IBST
       BACKSPACE IHT
   220 CONTINUE
       IREC=IBST-2
       KTRK=1
       DO 230 N=1,IREC
       READ(IHT) K,(ITSK(I),I=1,K)
       CALL DSKSTO (465,ITSK(1),KTRK,0)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          KTRK=KTRK+1
   230    CONTINUE
          READ(IHT) K,(IHVBUF(I),I=1,K)
          REWIND IHT
          DO 400 IJD=1,NJD
C
C         READ CONTROL,INDEX TITLE, AND TITLE CARDS AGAIN
          READ(ISCR)JEND,MVAR(IJD),LVAR(IJD),ISIZE(IJD),ISKL(IJD),
        1 ISEQ,IHEAD,IWT,IBCD(NJD),IDSP(NJD),ALO(NJD),AHI(NJD)
          READ(ISCR)(ILAB(I),I=1,8)
          READ(ISCR)((HEAD(I,K),I=1,14),K=1,IHEAD)
          IF(IBAD(IJD).NE.0) GO TO 400
C
C         COMPUTE MEANS, SD.
          DO 240 J=1,NTASK
          IF(ACNT(J,IJD).EQ.0.) GO TO 235
          SD(J,IJD)=SQRT((SUMX2(J,IJD)*ACNT(J,IJD)-SUMX(J,IJD)**2))/
        1 ACNT(J,IJD)
          MEAN(J,IJD)=SUMX(J,IJD)/ACNT(J,IJD)
   235    INX(J)=J
   240    CONTINUE
C
C         PUNCH MEANS IF WEIGHT DECK REQUESTED.
          IF(IWT.NE.0) GO TO 242
          WRITE(1,37) IDSF(IJD)
    37    FORMAT(A6)
          WRITE(1,38)
    38    FORMAT('(I2,A2,A4,10F7.2)')
          ICARD = 1
          IPT = 1
   241    KPT = IPT+9
          IF(KPT.GT.NTASK) KPT=NTASK
          WRITE(1,39) ICARD,ISID1,(MEAN(I,IJD),I=IPT,KPT)
    39    FORMAT(I2,'AV',A4,10F7.2)
          ICARD = ICARD + 1
          IPT = KPT + 1
          IF(KPT.NE.NTASK) GO TO 241
C
   242    GO TO(250,250,272),ISEQ
C
C         SORT ASCENDING/DESCENDING BY MEAN.
   250    IMORE=0
          DO 270 I=1,ITASK
          GO TO(260,265),ISEQ
   260    IF(MEAN(I,IJD).LE.MEAN(I+1,IJD)) GO TO 270
          GO TO 266
   265    IF(MEAN(I,IJD).GE.MEAN(I+1,IJD)) GO TO 270
   266    TEMP=MEAN(I,IJD)
          MEAN(I,IJD)=MEAN(I+1,IJD)
          MEAN(I+1,IJD)=TEMP
```

LINE #   TEXT...

```
            TEMP=SD(I,IJD)
            SD(I,IJD)=SD(I+1,IJD)
            SD(I+1,IJD)=TEMP
            TEMP=ACNT(I,IJD)
            ACNT(I,IJD)=ACNT(I+1,IJD)
            ACNT(I+1,IJD)=TEMP
            TEMP=INX(I)
            INX(I)=INX(I+1)
            INX(I+1)=TEMP
            IMORE=1
      270   CONTINUE
            IF(IMORE.EQ.0) GO TO 272
            GO TO 250
C
C           PRINT REPORT.  WRITE BCD OUTPUT TAPE IF REQUESTED.
      272   IPASS=IBCD(IJD)
            DO 382 KK=1,IPASS
            IPAGE=1
            KCNT = IHEAD+12
            LM=IOUT(KK)
            WRITE(LM,40) ((ILAB(IP),IP=1,8),IDSP(IJD),IPAGE
       40   FORMAT(1H1,8A6,29X,A6,6H PAGE ,I3,/)
            WRITE(LM,42)((HEAD(I,K),I=1,14),K=1,IHEAD)
       42   FORMAT(1X,13A6,A2)
            WRITE(LM,44) NMEM(IJD)
       44   FORMAT(34H NUMBER OF MEMBERS IN THE GROUP = ,I5,/,1H )
            WRITE(LM,46)
       46   FORMAT(/35X,57HSTANDARD DEVIATION OF VALUES FOR THOSE PERFORMING.
          1........,/35X,57HAVERAGE VALUE FOR THOSE PERFORMING................
          2        .,/35X,57HNUMBER PERFORMING........................      .
          3        .,/35X,57HSEQUENCE NUMBER.......................    .    .
          4        .,/72X,20H.    .       .        .,/7H D  TSK,28X,16HDUTY/TASK T
          5ITLES,21X,20H.    .       .         .,/1H )
            M=0
            DO 290 I=1,NTASK
            IF(ACNT(I,IJD).LE.AMIN) GO TO 290
            M=M+1
            J=INX(I)
            KTRK=IHVBUF(J)/INUM(3)
            ILOC = MOD(IHVBUF(J),INUM(3))
            CALL DSKRED(465,ITSK(1),KTRK,0)
            ISTART=ILOC+1
            ISTOP=ISTART+10
            IRCDS=(ITSK(ILOC)/INUM(3)-(ITSK(ILOC)-ITSK(ILOC)/INUM(3)*INUM(3)
          1))/11
            ILINES=IRCDS
            JCNT=ACNT(I,IJD)
            WRITE(LM,57)(ITSK(IX),IX=ISTART,ISTOP),M,JCNT,MEAN(I,IJD),
          1 SD(I,IJD)
       57   FORMAT(1H ,11A6,3X,I3,I5,F8.2,F8.2)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
  278 IF(IRCDS.EQ.1) GO TO 280
      IRCDS=IRCDS-1
      ISTART=ISTART+11
      ISTOP=ISTOP+11
      WRITE(LM,59)(ITSK(IX),IX=ISTART,ISTOP)
   59 FORMAT(1H ,11A6)
      GO TO 278
  280 KCNT=KCNT+ILINES
      IF(KCNT.LT.55) GO TO 290
     IF(I.EQ.NTASK) GO TO 382
      KCNT=2
      IPAGE = IPAGE+1
      WRITE(LM,40) (ILAB(IP),IP=1,8),IDSP(IJD),IPAGE
  290 CONTINUE
  382 CONTINUE
C
C     PUNCH REQUEST CARD IF REPORT WRITTEN ON BCD TAPE.
  385 IF(IBCD(IJD).NE.2) GO TO 400
      CALL REPEND(ISID1,IDSP(IJD),ILAB(1),JSEQ)
  400 CONTINUE
C
C     GET SOME MORE CONTROL CARDS IF NO END CARD ENCOUNTERED.
  405 IF(LEND.EQ.IEND) GO TO 407
      READ(IMT) K
      GO TO 107
  407 RETURN
      END
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

```
LINE #   TEXT...

      C         INTRODUCTION
      C
      C SECTION I     = STATEMENT NUMBERS    1 TO  99
      C                 SUBJECT = CONTROL CARDS AND PRE-CASE DATA ON TAPE
      C
      C SECTION II    = STATEMENT NUMBERS  100 TO 199
      C                 SUBJECT = CASE DATA LOOP WITH DISTRIBUTION
      C
      C SECTION III   = STATEMENT NUMBERS  200 TO 299
      C                 SUBJECT = COMPUTATION OF TASK MEANS AND STANDARD DEV.
      C
      C SECTION IV    = STATEMENT NUMBERS  300 TO 349
      C                 SUBJECT = COPYING TASK TITLES TO DISK (FIRST PASS ONLY)
      C
      C SECTION V     = STATEMENT NUMBERS  350 TO 399
      C                 SUBJECT = PUNCH THEN SORT IF REQUIRED
      C
      C SECTION VI    = STATEMENT NUMBERS  400 TO 449
      C                 SUBJECT = BEGIN PRINT SECTION CONTROL LOOP
      C
      C SECTION VII   = STATEMENT NUMBERS  600 TO 699
      C                 SUBJECT = PRINT SECTION
      C
      C SECTION VIII  = STATEMENT NUMBERS  450 TO 499
      C                 SUBJECT = END OF PRINT SECTION CONTROL LOOP
      C
      C SECTION IX    = STATEMENT NUMBERS  700 TO 799
      C                 SUBJECT = RESET AND RECYCLE
      C
      C SECTION X     = STATEMENT NUMBERS  900 TO 999
      C                 SUBJECT = ERROR MESSAGES, END OF MEDIA, AND STOP
      C
            INTEGER V, BLANK, ONE, END, A, S, TSKLEV, HIST, TASK
      C
            DATA V,BLANK,ONE,END,A,S,IOTHER/'V',' ','1','END','A','S','OTHER'/
            DATA ITOTAL/ 'TOTAL'/
            DATA ASTER/6H******/
            DATA IEND1/6HEND OF/
      C
            DIMENSION IHEAD(8), ITIT(14,3), LOCTSK(1726), LIST(75), ITYPE(11),
           1MEM(11), IVAL(9), IFILL(4), IVAR(9), LOGIC(8), ITOP(2,9),
           2XTOP(2,1), IBOT(2,9), XBOT(2,1), LENGTH(9),  LOCHIS(1000), C(500),
           3IC(1), KTIT(465), LTIT(11,9), HIST(1000), TMN(1726), TSD(1726),
           4TASK(1726), TSKLEV(870,9), CORE(15311), IFIND(1726)
            DIMENSION M(1726)
      C
            EQUIVALENCE (CORE,IHEAD),(CORE(9),ITIT),(CORE(61),LOCHIS),
           1(CORE(1061),LIST,ITYPE),(CORE(1072),MEM),(CORE(1083),IVAL),
           2(CORE(1135),IFILL),(CORE(1139),IVAR),(CORE(1148),LOGIC),
           3(CORE(1156),ITOP,XTOP),(CORE(1174),IBOT,XBOT),(CORE(1192),LENGTH),
```

```
LINE #   TEXT...

          4(CORE(1210),LTIT,C,IC),      (CORE(1710),HIST,TMN),(J1,V1),(J2,V2),
          5(CORE(3460),TASK,TSD),(CORE(5210),TSKLEV),(CORE(13062),LOCTSK),
          6(CORE(5210),IFIND), (CORE(6910),KTIT)
      C
          ASSIGN 300 TO IDISK
          ASSIGN 50 TO ITAPE
      C
          IDSAM = 1
          ISHF18 = 2**18
      C
          IRUNS = 0
          IGOOD = 0
          IREEL = 1
          CALL SETUP(10,26,0,0,0,0,20,0,0,0,0,JSEQ)
          IP = 10
          IS = 20
          IR = 26
          CALL RESET
      C
      C SECTION I     = STATEMENT NUMBERS    1 TO  99
      C                 SUBJECT = CONTROL CARDS AND PRE-CASE DATA ON TAPE
      C
      C  READ PROGRAM AND TAPE IDENTIFICATION CARD
      C
          READ(5,800) ID1, ID2
          CALL PRINTC
      C
      C  PROGRAM RE-CYCLE POINT
      C
  777     CONTINUE
          ASSIGN 109 TO ISAMP
          KTYP = 1
          IPASS = 1
          IABSL = 8
          IPAGE = 1
          NVAR = 0
          REWIND IP
      C
      C  CORE STORAGE IS RESET TO ZERO/BLANKS
          CALL ZEROI(TSKLEV,7830)
          CALL ZEROI(M,1726)
      C
      C  READ CONTROL CARD (IF END OF FILE, GO TO 999)
      C
          READ(5,801,END=999) KODE,LV,ID,NTIT,LSEQ,JPUN,IHEAD,JBCD,IDREPT
          CALL PRINTC
          IRUNS = IRUNS + 1
          IFEND = 0
      C
          ASSIGN 108 TO JSKIP
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
        IF(KODE.EQ.A) ASSIGN 106 TO JSKIP
        ASSIGN 150 TO LSKIP
        IF(KODE.EQ.A.OR.KODE.EQ.S) ASSIGN 110 TO LSKIP
C
        IF(LV.NE.V) GO TO 901
        CALL BCDBIN(ID,ID,3,$902)
C DETERMINE THE NUMBER OF TITLE CARDS
        CALL BCDBIN(NTIT,NTIT,1,$903)
C IF NOT WITHIN LIMITS, GO TO ERROR ROUTINE
        IF(NTIT.LT.0.OR.NTIT.GT.3) GO TO 903
C
        ISEQ = 0
        IPUN = 0
        IBCD = 2
        IF(LSEQ.EQ.ONE) ISEQ = 1
        IF(JPUN.EQ.ONE) IPUN = 1
        IF(JBCD.EQ.ONE) IBCD = 1
C
        IF(NTIT.EQ.0) GO TO 3
C
C IF NUMBER OF TITLE CARDS IS NON-ZERO, READ THEM IN
        DO 2 J = 1, NTIT
        READ(5,802) (ITIT(I,J),I=1,14)
        CALL PRINTC
2       CONTINUE
C
3       CONTINUE
C
C READ NEXT CARD, IF END CARD, SKIP FOLLOWING SECTION
        IPRNT = 1
        IERR = 0
C PROCESS VARIABLE INTERACTION CARD
        CALL SAMPLE(IDSAM,IVAR,NVAR,IERR,IPRNT)
        IF(IERR.NE.-1) GO TO 77542
        IERR = 0
        IFEND = 'END'
        GO TO 9
77542 CONTINUE
C
C SAMPLE SELECTION SECTION
C
        ASSIGN 100 TO ISAMP
C
        IF(NVAR.LT.1.OR.NVAR.GT.9) GO TO 904
C
        IF(IERR.NE.0) GO TO 905
C READ ONE LIMIT CARD FOR EACH VARIABLE
C NOTE - THESE MUST BE INPUT IN THE SAME ORDER AS THEY APPEARED
C         ON THE VARIABLE INTERACTION CARD
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      DO 8 I = 1, NVAR
      ISUB = IVAR(I)
      IF(ISUB.LT.0) GO TO 6
      IF(ISUB.GT.3) GO TO 5
C
      READ(5,804) IFEND, JT, J1, J2
      CALL PRINTC
      IF(IFEND.EQ.END) GO TO 909
      CALL BCDBIN(JT,JT,3,$902)
      IF(JT.NE.ISUB) GO TO 906
      CALL BCDBIN(J1,J1,6,$907)
      CALL BCDBIN(J2,J2,6,$907)
      ITOP(1,I) = J2
      IBOT(1,I) = J1
      LENGTH(I) = 1
      GO TO 8
C
5     CONTINUE
      READ(5,806) IFEND, JT, V1, V2
      CALL PRINTC
      IF(IFEND.EQ.END) GO TO 909
      CALL BCDBIN(JT,JT,3,$902)
      IF(JT.NE.ISUB) GO TO 906
      XTOP(1,I) = V2
      XBOT(1,I) = V1
      GO TO 8
C
6     CONTINUE
      READ(5,804) IFEND, JT, J1, J2, J3, J4
      CALL PRINTC
      IF(IFEND.EQ.END) GO TO 909
      CALL BCDBIN(JT,JT,3,$902)
      JT = -(JT)
      IF(JT.NE.ISUB) GO TO 906
      LHIS = IFEND
      CALL BCDBIN(LHIS,LHIS,2,$902)
      IF(LHIS.EQ.0) LHIS = 1
      IF(LHIS.EQ.1) GO TO 7
      IF(LHIS.NE.2) GO TO 907
      ITOP(1,I) = J3
      ITOP(2,I) = J4
      IBOT(1,I) = J1
      IBOT(2,I) = J2
      LENGTH(I) = 2
      GO TO 8
C
7     CONTINUE
      ITOP(1,I) = J2
      IBOT(1,I) = J1
      LENGTH(I) = 1
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

```
LINE #   TEXT...

      8     CONTINUE
      C
      C  READ NEXT CARD, IF COUNTS ARE CORRECT, IT IS AN END CARD
      C
            READ(5,807,END=999) IFEND
            CALL PRINTC
            IF(IFEND.NE.END) GO TO 910
      9     CONTINUE
      C
      C
      C
      C  BEGIN TO PROCESS THE INPUT TAPE
      C
      C
      C  READ THE COMMUNICATIONS REGION ON THE FIRST PASS
      C
            GO TO ITAPE, (50,52)
      50    ASSIGN 52 TO ITAPE
      C
      C  READ THE COMMUNICATIONS REGION
            READ(IP) N, (IC(I),I=1,N)
            IF(ID1.NE.IC(1).OR.ID2.NE.IC(2)) GO TO 913
            NCASE = IC(3)
            NDICT = IC(4)
            NDUTY = IC(9)
            NTASK = IC(12)
            CALL DISK(NTASK,99)
      C
      C  READ RELATIVE LOCATION TABLE
            READ(IP) N, (LOCHIS(I),I=1,N)
      C
      C  BYPASS DICTIONARY BLOCKS
            DO 51 I = 1, NDICT
      51    READ(IP)
            JPASS = NDICT + 2
            GO TO 54
      C
      C  ON SECOND AND SUBSEQUENT PASSES, SKIP TO CASE DATA
      52    REWIND IP
            DO 53 I = 1, JPASS
      53    READ(IP)
      C
      54    CONTINUE
      C
      C
      C  CONVERT IVAR OF VXXX TO (-RELATIVE LOCATIONS)
            IF(NVAR.EQ.0) GO TO 76
      C
            DO 75 I = 1, NVAR
            ISUB = IVAR(I)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      IF(ISUB.GE.0) GO TO 75
      ISUB = -(ISUB)
      IVAR(I) = -(MOD(LOCHIS(ISUB),ISHF18))
75    CONTINUE
C
76    ID = MOD(LOCHIS(ID),ISHF18)
      DO 99 I = 1, 11
      ITYPE(I) = 1
99    MEM(I) = 0
      ITYPE(10) = IOTHER
      ITYPE(11) = ITOTAL
C
C
C
C SECTION II   = STATEMENT NUMBERS  100 TO 199
C                SUBJECT = CASE DATA LOOP WITH DISTRIBUTION
C
C
C  BEGIN CASE DATA LOOP
C
      DO 199 ICASE = 1, NCASE
C
C  READ NEXT CASE
      READ(IP) NC,(C(I),I=1,NC),NH,(HIST(I),I=1,NH),NT,(TASK(I),I=1,NT)
C
      GO TO ISAMP, (100,109)
C
C
C
C  RANGE CHECK EVERY VARIABLE IN SAMPLE SELECTION
C
100   CONTINUE
C
      DO 104 I = 1, NVAR
      IVAL(I) = 0
      ISUB = IVAR(I)
      IF(ISUB.LT.0) GO TO 102
      IF(ISUB.LT.3) GO TO 101
C
      IF(C(ISUB).GE.XBOT(1,I).AND.C(ISUB).LE.XTOP(1,I)) IVAL(I) = 1
      GO TO 104
C
101   IF(IC(ISUB).GE.IBOT(1,I).AND.IC(ISUB).LE.ITOP(1,I)) IVAL(I) = 1
      GO TO 104
C
102   ISUB = -(ISUB)
      L = LENGTH(I)
C  USE A LOGICAL ALPHABETIC COMPARE ON VXXX VARIABLES
      IVAL(I) = LASCMP(HIST(ISUB),IBOT(1,I),ITOP(1,I),L)
104   CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C  DETERMINE IF THIS CASE IS TO BE KEPT
C      TO USE CASE, CONTINUE AT 109
C      TO BYPASS CASE, GO TO 199
       IERR = 0
       CALL SELECT(IDSAM,IVAL,$199,$199)
C
C  END SAMPLE SELECTION SECTION
C
C  PICK OFF PROPER LEVEL CHARACTER
C
 109   CONTINUE
       GO TO JSKIP, (106,108)
 105   MEM(10) = MEM(10) + 1
       GO TO 199
 106   DECODE(107,HIST(ID)) LEVEL
 107   FORMAT(3X,A1)
       GO TO 111
 108   LEVEL = HIST(ID)
 111   CALL BCDBIN(LEVEL,LEVEL,1,$105)
       GO TO LSKIP, (110,150)
 110   IF(MOD(LEVEL,2).EQ.0) GO TO 105
C
 150   MEM(LEVEL) = MEM(LEVEL) + 1
       MEM(11) = MEM(11) + 1
C
       DO 160 INDEX = 1, NT
       ITASK =  TASK(INDEX) / ISHF18
       M(ITASK) = M(ITASK) + 1
       LTASK = (ITASK-1)/2
       MTASK = 0
       IF(MOD(ITASK,2).EQ.0) MTASK = 18
       N = FLD(MTASK,18,TSKLEV(LTASK,LEVEL))
       FLD(MTASK,18,TSKLEV(LTASK,LEVEL)) = N + 1
 160   CONTINUE
 199   CONTINUE
C
C  END CASE DATA LOOP
C
C
C SECTION III  = STATEMENT NUMBERS  200 TO 299
C                SUBJECT = COMPUTATION OF TASK MEANS AND STANDARD DEV.
C
C
C
       IF(MEM(11).EQ.0) GO TO 914
       MINUS = 2
       IF(MEM(10).EQ.0) MINUS = 1
       I = 0
       DO 200 J = 1, 11
       IF(MEM(J).EQ.0) GO TO 200
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
         I = I + 1
         MEM(I) = MEM(J)
         ITYPE(I) = ITYPE(J)
  200    CONTINUE
         NTYP2 = I
         NTYP = I - MINUS
C
C
C   SUMMARY OF AVGPCT CALCULATIONS
C
C        TOT = NUMBER OF MEMBERS IN THIS LEVEL
C        PRF = NUMBER OF MEMBERS IN THIS LEVEL PERFORMING THIS TASK
C        PCTPRF =  PRF / TOT * 100.0 = PERCENT PERFORMING THIS TASK
C        F = FREQUENCY GIVEN BY THE SUMS OF PCTPRF FOR EACH LEVEL
C        X = THE LEVEL (MAY BE GRADE, SKILL LEVEL, ETC.)
C        SUMFX = SUMMATIONS OF PCTPRF TIMES LEVEL FOR EACH TASK
C        SFXSQ = SUMMATIONS OF PCTPRF TIMES LEVEL SQUARED FOR EACH TASK
C        TMN(ITASK) = SUMFX / F   = AVGPCT TASK MEAN
C        TSD(ITASK) = SQUARE ROOT OF(SFXSQ/F - TMN*TMN) = AVGPCT TASK S.D.
         DO 210 ITASK = 1, NTASK
         F = 0.0
         SUMFX = 0.0
         SFXSQ = 0.0
C
         DO 205 INDEX = 1, NTYP
         LEVEL = ITYPE(INDEX)
         X = LEVEL
         TOT = MEM(INDEX)
         LTASK = (ITASK-1)/2
         MTASK = 0
         IF(MOD(ITASK,2).EQ.0) MTASK = 18
         N = FLD(MTASK,18,TSKLEV(LTASK,LEVEL))
         PRF = N
         PCTPRF = PRF / TOT * 100.0
         F = F + PCTPRF
         SUMFX = SUMFX + PCTPRF * X
         SFXSQ = SFXSQ + PCTPRF * X * X
  205    CONTINUE
C
         TM = SUMFX / F
         TMN(ITASK) = TM
         TSD(ITASK) = SQRT(ABS(SFXSQ/F - TM * TM))
  210    CONTINUE
C
C
C SECTION IV    = STATEMENT NUMBERS  300 TO 349
C                 SUBJECT = COPYING TASK TITLES TO DISK (FIRST PASS ONLY)
C
C
C  BYPASS DISK LOADER AFTER FIRST PASS
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

AVGPCT- AVERAGE BY PER CENT MEMBERS PERFORMING

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C
      GO TO IDISK(300,350)
300   ASSIGN 350 TO IDISK
C
C  CHECK CASE E O D
C
      READ(IP) N, LENDD
      IF(LENDD.NE.IENDI) GO TO 335
C
C  TRANSFER TITLES AND LOCATORS TO DISK
C
      REWIND IS
      KTREC = 0
304   KTREC = KTREC + 1
      READ(IP) NWIR, (KTIT(JK),JK=1,NWIR)
      WRITE(IS)NWIR, (KTIT(JK),JK=1,NWIR)
      IF(KTIT(1).NE.IENDI) GO TO 304
      REWIND IS
C
C  MOVE TO BEGINNING OF TITLE LOCATOR RECORD
C
      BACKSPACE IP
      BACKSPACE IP
      BACKSPACE IP
C
C  PUT LOCTSK TITLE LOCATOR INTO CORE
C
      READ(IP) N, (LOCTSK(JK),JK=1,N)
C
C
C  PUT TITLES ONTO DISK, ONE TITLE PER TRACK
C
C
      IREC = 0
      DO 330 ITASK = 1,NTASK
      JREC = LOCTSK(ITASK) / ISHF18
C
C  JREC CONTAINS RECORD NUMBER OF 'ITASK-TH' TITLE
C
      IF(IREC.EQ.JREC) GO TO 310
C
C  READ RECORD INTO CORE
C
      READ(IS) N, (KTIT(JK),JK=1,N)
      IREC = JREC
C
310   NSCW = MOD(LOCTSK(ITASK),ISHF18)
      NSLW = KTIT(NSCW) / ISHF18
      NW = NSLW - NSCW
      NSCW = NSCW + 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      LOCTSK(ITASK) = NW
C
C  STORE TITLE ON DISK TRACK
C
      CALL DSKSTD(NW,KTIT(NSCW),ITASK,0)
330   CONTINUE
      GO TO 350
335   CONTINUE
      WRITE(6,867)
      STOP
350   CONTINUE
C
C
C SECTION V     = STATEMENT NUMBERS  350 TO 399
C                 SUBJECT = PUNCH THEN SORT IF REQUIRED
C
C
C
      IF(IPUN.NE.1) GO TO 375
C
      NCARD = (NTASK - 1) / 14 + 1
      I = 0
      K = 0
      IPNCH = 1
      WRITE(IPNCH,809)
      DO 360 II = 1, NCARD
      I = K + 1
      K = K + 14
      IF(K.GT.NTASK) K = NTASK
      WRITE(IPNCH,808) II, IDI, (TMN(L),L=I,K)
360   CONTINUE
C
375   CONTINUE
C
      DO 376 I = 1, NTASK
376   IFIND(I) = I
      IF(ISEQ.NE.1) GO TO 399
C
C     IF SORT REQUIRED, PERFORM A BUBBLE SORT FOR TMN DESCENDING SEQ.
C
      DO 390 K = 2, NTASK
      I = K - 1
      J = K
C
C  IF THIS VALUE IS LESS THAN OR EQUAL TO PREVIOUS VALUE, ORDER IS OK
380   IF(TMN(J).LE.TMN(I)) GO TO 390
C
C  IF THE VALUE EXCEEDS THE PREVIOUS ONE, SWITCH THEIR POSITIONS
      T1 = TMN(I)
      T2 = TSD(I)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      K1 = IFIND(I)
      K2 = M(I)
      TMN(I) = TMN(J)
      TSD(I) = TSD(J)
      IFIND(I) = IFIND(J)
      M(I) = M(J)
      TMN(J) = T1
      TSD(J) = T2
      IFIND(J) = K1
      M(J) = K2
      J = I
      I = I - 1
C
C  CHECK TO SEE IF THIS VALUE EXCEEDS THE PRIOR VALUE
      IF(I.GT.0) GO TO 380
C
C
390   CONTINUE
C
C
399   CONTINUE
C
C
C
C
C SECTION VI    = STATEMENT NUMBERS  400 TO 449
C                 SUBJECT = BEGIN PRINT SECTION CONTROL LOOP
C
      WRITE(6,880) IHEAD,IDREPT,IPAGE
      IF(IBCD.EQ.2) WRITE(IR,880) IHEAD, IDREPT, IPAGE
      IF(NTIT.EQ.0) GO TO 603
C
      DO 602 ILOOP = 1,NTIT
      IF(IBCD.EQ.2) WRITE(IR,882) (ITIT(KWD,ILOOP),KWD=1,14)
602   WRITE(6,882) (ITIT(KWD,ILOOP),KWD = 1,14)
C
603   WRITE(6,881)
      IF(IBCD.EQ.2) WRITE(IR,881)
C
      JTASK = 0
C
      DO 499 ITASK = 1, NTASK
      IF(TMN(ITASK).EQ.0.0) GO TO 499
      JTASK = JTASK + 1
      ISUB = IFIND(ITASK)
      LENG = LOCTSK(ISUB)
C
C  RETREIVE PROPER TASK TITLE FROM DISK (STORE IN LTIT)
C
      CALL DSKRED(LENG,LTIT,ISUB,0)
```

LINE #   TEXT...

```
        NL = (LENG - 1) / 11 + 1
C SECTION VII  = STATEMENT NUMBERS  600 TO 699
C                SUBJECT = PRINT SECTION
C
        LTYP = KTYP
        LPASS = IPASS
C
C       CHECK FOR PAGE OVERRUN
C
        IF(IABSL + NL .LE. IPAGE * 53) GO TO 605
C
C       PAGE HEADER
C
        IABSL = IPAGE * 53
        IPAGE = IPAGE + 1
C
C
  604   WRITE(6,850) IHEAD,IDREPT,IPAGE
        IF(IBCD.EQ.2) WRITE(IR,850) IHEAD, IDREPT, IPAGE
C
C
C       EXECUTE OUTPUT LOOP
C
  605   DO 690 J = 1,IBCD
        KTR = 1
C
C       CHOOSE OUTPUT DEVICE
C
        IOUT = 6
        IF(J.EQ.1) GO TO 606
        IOUT = IR
        KTYP = LTYP
        IPASS = LPASS
C
C       BRANCH FOR VALUE-MEMBERS DISPLAY
C
  606   IF(IPASS.EQ.1) GO TO 660
C
C       FIRST TITLE LINE
C
        WRITE (IOUT,883) (LTIT(NWORD,KTR),NWORD = 1,11),JTASK,TMN(ITASK),
       1TSD(ITASK), M(ITASK)
        KTR = KTR+1
        IF(KTR.GT.NL) GO TO 689
C
C       CONTINUATION LINE(S)
C
  609   DO 610 INDEX = KTR,NL
        WRITE (IOUT,884) (LTIT(NWORD,INDEX),NWORD = 1,11)
  610   CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          GO TO 689
C
C
C         VALUE-MEMBERS DISPLAY SECTION
C
C         IF READY FOR NEXT TITLE...
  650     IF (KTR.GT.NL) GO TO 689
C
C         IF ON FIRST LINE OF TITLE...
          IF (KTR.EQ.1) GO TO 660
C
C         CONTINUATION LINE(S) AND VALUE-MEMBERS DISPLAY
C
          IF(KTYP.GT.NTYP2) GO TO 658
          IF (KTYP.GT.NTYP) GO TO 656
          WRITE (IOUT,854) (LTIT(NWORD,KTR),NWORD = 1,11),ITYPE(KTYP),MEM(KT
         1YP)
          GO TO 668
  656     WRITE (IOUT,864) (LTIT(NWORD,KTR),NWORD = 1,11),ITYPE(KTYP),MEM(KT
         1YP)
          GO TO 668
  658     WRITE (IOUT,894) (LTIT(NWORD,KTR),NWORD = 1,11),ASTER,ASTER
          GO TO 671
C
C         FIRST TITLE LINE AND VALUE-MEMBERS DISPLAY
C
  660     IF(KTYP.GT.NTYP2) GO TO 666
          IF (KTYP.GT.NTYP) GO TO 664
          WRITE (IOUT,853) (LTIT(NWORD,KTR),NWORD = 1,11),JTASK,TMN(ITASK),
         1TSD(ITASK), M(ITASK), ITYPE(KTYP), MEM(KTYP)
          GO TO 668
  664     WRITE (IOUT,863) (LTIT(NWORD,KTR),NWORD = 1,11),JTASK,TMN(ITASK),
         1TSD(ITASK), M(ITASK), ITYPE(KTYP), MEM(KTYP)
          GO TO 668
  666     WRITE (IOUT,893) (LTIT(NWORD,KTR),NWORD = 1,11),JTASK,TMN(ITASK),
         1TSD(ITASK), M(ITASK), ASTER, ASTER
          GO TO 671
  668     CONTINUE
          KTYP = KTYP+1
          KTR = KTR +1
          GO TO 650
  671     CONTINUE
          KTR = KTR +1
C
C         SET VALUE-MEMBERS-DISPLAY BYPASS SWITCH
C
          IPASS = 2
          IF (KTR.GT.NL) GO TO 689
          GO TO 609
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C      THE RARELY-INVOKED VALUE-MEMBERS DISPLAY OVERRUN SECTION
C
675    IF (IPASS.NE.1) GO TO 698
680    IF(KTYP.GT.NTYP2) GO TO 684
       IF (KTYP.GT.NTYP) GO TO 682
       WRITE (IOUT,871) ITYPE(KTYP),MEM(KTYP)
       GO TO 687
682    WRITE (IOUT,872) ITYPE(KTYP),MEM(KTYP)
       GO TO 687
684    WRITE (IOUT,873) ASTER,ASTER
       GO TO 698
687    KTYP = KTYP+1
       GO TO 680
689    CONTINUE
C
C      CHECK FOR END OF TITLE LIST
C
       IF (ITASK.EQ.NTASK) GO TO 675
690    CONTINUE
C
       IABSL = IABSL + NL
C
C
698    CONTINUE
C
C
C SECTION VIII = STATEMENT NUMBERS  450 TO 499
C                SUBJECT = END OF PRINT SECTION CONTROL LOOP
C
C
499    CONTINUE
C
       IF(IBCD.EQ.2) CALL REPEND(ID1,IOREPT,IHEAD,JSEQ)
C
       IGOOD = IGOOD + 1
C
C
C SECTION IX   = STATEMENT NUMBERS  700 TO 799
C                SUBJECT = RESET AND RECYCLE
C
C
700    CALL RESET
       GO TO 777
800    FORMAT(7X,A6,A2)
801    FORMAT(4X,A1,1X,A1,A3,1X,A1,1X,A1,1X,A1,1X,8A6,1X,A1,1X,A6)
802    FORMAT(13A6,A2)
803    FORMAT(A3,76A1)
804    FORMAT(A3,1X,A3,1X,4A6)
806    FORMAT(A3,1X,A3,1X,2F12.0)
```

LINE #   TEXT...

```
 807   FORMAT(A3)
 808   FORMAT(I2,'AP',A4,14F5.2)
 809   FORMAT('(I2,A2,A4,14F5.2)')
 850   FORMAT(1H1,8A6,29X,A6,' PAGE ',I4,//7H D  TSK,28X,16HDUTY/TASK TIT
      1LES,/)
 853   FORMAT(1X,11A6,I7,2F8.2,I7,10X,'* ',I5,I11,' *')
 854   FORMAT(1X,11A6,40X,'* ',I5,I11,' *')
 863   FORMAT(1X,11A6,I7,2F8.2,I7,10X,'* ',A5,I11,' *')
 864   FORMAT(1X,11A6,40X,'* ',A5,I11,' *')
 867   FORMAT('1FIRST EOD NOT FOUND...ASSUME NCASE IS INCORRECT')
 871   FORMAT(107X,'* ',I5,I11,' *')
 872   FORMAT(107X,'* ',A5,I11,' *')
 873   FORMAT(107X,'*****',2A6,'*****')
 880   FORMAT(1H1,8A6,29X,A6,' PAGE ',I4,//)
 881   FORMAT(1H0,35X,'NUMBER PERFORMING',44(1H.),/36X,'STANDARD DEVIATIO
      1N OF VALUES FOR THOSE PERFORMING...',8X,1H.,/36X,'AVERAGE VALUE FO
      2R THOSE PERFORMING',10(1H.),7X,1H.,8X,1H.,/36X,'SEQUENCE NUMBER',
      323(1H.),5X,1H.,7X,1H.,8X,1H.,/73X,1H.,5X,1H.,7X,1H.,8X,1H.,10X,
      420(1H*),/,' D  TSK',29X,'DUTY/TASK TITLES',21X,1H.,5X,1H.,7X,1H.,
      58X,1H.,10X,'* VALUE    MEMBERS *',/107X,'*',18X,'*')
 882   FORMAT(1X,13A6,A2)
 883   FORMAT(1X,11A6,I7,2F8.2,I7)
 884   FORMAT(1X,11A6)
 893   FORMAT(1X,11A6,I7,2F8.2,I7,10X,'*****',2A6,'*****')
 894   FORMAT(1X,11A6,40X,'*****',2A6,'*****')
C
C
C
C SECTION X     = STATEMENT NUMBERS  900 TO 999
C                 SUBJECT = ERROR MESSAGES, END OF MEDIA, AND STOP
C
 900   CALL ERRMSG(8,'** SEARCHING FOR END CARD DUE TO ABOVE ERROR    ')
       IF(IFEND.EQ.END) GO TO 700
 925   READ(5,807) IFEND
       CALL PRINTC
       IF (IFEND.NE.END) GO TO 925
       GO TO 700
 901   CALL ERRMSG(8,'** AVGPCT CAN AVERAGE ONLY V VARIABLES - SEE CC7')
       GO TO 900
 902   CALL ERRMSG(8,'** ABOVE CARD HAS INVALID VARIABLE ID          ')
       GO TO 900
 903   CALL ERRMSG(8,'** NUMBER OF TITLE CARDS IS NOT IN RANGE (0 - 3)')
       GO TO 900
 904   CALL ERRMSG(8,'** NUMBER OF VARIABLES IS NOT IN RANGE (1 - 9)  ')
       GO TO 900
 905   CALL ERRMSG(8,'** INCORRECT VARIABLE ID OR OPERATOR IN CC 4-79 ')
       GO TO 900
 906   CALL ERRMSG(8,'** VARIABLE ID DOES NOT MATCH CORRESPONDING ID  ')
       GO TO 900
 907   CALL ERRMSG(8,'** INVALID INTEGER ABOVE - SEE CC  9-14 + 15-20 ')
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          GO TO 900
    909   CALL ERRMSG(8,'** UNEXPECTED END CARD - LIMITS CARD MISSING     ')
          GO TO 700
    910   CALL ERRMSG(8,'** END CARD IS MISSING - EXTRA CARDS PRESENT     ')
          GO TO 900
    C
    913   CALL ERRMSG(8,'** TAPE ID ON CARD DOES NOT MATCH TAPE ID        ')
          ID1 = IC(1)
          ID2 = IC(2)
          GO TO 1000
    914   CALL ERRMSG(8,'** NO VALID CASES FOUND IN THE SAMPLE ON TAPE    ')
          GO TO 700
    C
    999   IF(IFEND.EQ.END) GO TO 1000
    911   CALL ERRMSG(8,'** UNEXPECTED END OF FILE ON CARD READER         ')
    1000  WRITE(6,920) ID1, ID2, IGOOD, IRUNS
    920   FORMAT('0END OF AVGPCT ON ',A6,A2,1X,I3,' OF',I3,' RUNS ARE GOOD')
          STOP
          END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),
     1          IVAR(100),TITLE(14),JA(50),IREL(1000),ICOMP(100),
     2          LHIST(100),TYPE(100),IVAL(100),IFMT(100),AA(1000),
     3          KB(1000),ARRAY(100),KC(15,20),A(11000),KA(11000),
     4          IVNUM(100),IBAD(100),ID(2)
      COMMON FMT1,FMT2,A
      EQUIVALENCE(FMT1,KFMT1),(FMT2,KFMT2),(A,KA), (A(1),IVAR(1)),
     1           (A(101),TITLE(1)), (A(115),JA(1)), (A(165),IREL(1)),
     2           (A(1165),ICOMP(1)), (A(1265),LHIST(1)),
     3           (A(1365),TYPE(1)), (A(1465),IVAL(1)),
     4           (A(1565),IFMT(1)),
     5           (A(1725),KB(1)), (A(2725),ARRAY(1)),
     6           (A(2825),KC(1,1)), (A(3125),IVNUM(1)),
     7           (A(3225),IBAD(1)),(A(3325),AA(1))
      DATA CORR/6HCORR  /, END/6HEND OF/
      DATA C/6HC     /,V/6HV     /
      DATA ICEE, IONE, ISIX / 1RC, 1R1, 1R6/
      ISHF6=2**6
      ISHF12=2**12
      ISHF18=2**18
       ISHF30=2**30
       IHT=10
       IOUT=15
       ISCR=20
       ISCR1=21
       CALL SETUP(IHT,IOUT,0,0,0,0,ISCR,ISCR1,0,0,0,JSEQ)
      CALL MARGIN
C****
C**** NC IS THE NUMBER OF COMPUTED VARIABLE DEFINITIONS READ FROM THE
C**** CODAP HISTORY TAPE.  NV IS THE NUMBER OF HISTORY VARIABLE
C**** DEFINITIONS READ.  NHIST IS THE NUMBER OF HISTORY VARIABLES
C**** SELECTED FOR CORRELATION.  NCOMP IS THE NUMBER OF COMPUTED
C**** VARIABLES SELECTED.  NCASE IS THE NUMBER OF CASES ON THE CODAP
C**** TAPE MINUS ALL CASES WHICH HAD AN OUT OF RANGE VARIABLE.  AN OUT
C**** OF RANGE VARIABLE IS ONE WHICH CONTAINS ALPHA CHARACTERS OR AN
C**** EMBEDDED BLANK.
      NC=0
      NV=0
      NHIST=0
      NCOMP=0
      NCASE=0
C****
C**** READ PROGRAM ID CARD.
      READ 2,IPROG,ID1,ID2,ITIT
    2 FORMAT(A6,X,A6,A2,I2)
      ILINE=ITIT+3
      IF(ITIT.EQ.0) GO TO 100
      PRINT 3
    3 FORMAT('1')
C****
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C***** READ AND PRINT TITLE CARDS.
       DO 90 J=1,ITIT
       READ 4,(TITLE(I),I=1,14)
       PRINT 5,(TITLE(I),I=1, 14 )
    90 CONTINUE
     4 FORMAT(13A6,A2)
     5 FORMAT(' ',14A6)
C*****
C***** READ A CARD.  IT WILL BE 'CORR' OR 'REGR'.
   100 READ 6, ANAME,NVAR,ID(1),ID(2),XVAR,IXVAR,FGOOD
       IGOOD = FGOOD
       NJUMP = 1
       IF(IXVAR.GT.0) NJUMP=2
     6 FORMAT(A4,I3,1X,A6,A2,1X,A1,I3,F2.0)
       IF(ANAME.NE. CORR ) GO TO 290
       IF(NVAR.LT.1.OR.NVAR.GT.100) GO TO 330
C*****
C***** A 'CORR CARD WAS READ.  READ COMMUNICATION REGION AND CHECK
C***** STUDY ID.
        READ(IMT) K,(JA(I),I=1,K)
       LCASE=JA(3)
       NDICT=JA(4)
       IF(JA(1).NE.ID1.OR.JA(2).NE.ID2) GO TO 300
C*****
C***** READ RELATIVE LOCATION TABLE AND SAVE.
        READ(IMT) K,(IREL(I),I=1,K)
C*****
C***** READ VARIABLE SPEC CARDS.  VARIABLES CAN BE READ IN ANY ORDER,
C***** BUT WILL BE SORTED IN ASCENDING ORDER.  VARIABLE DEFINITIONS WILL
C***** BE PRINTED IN ASCENDING ORDER, COMPUTED VARIABLES FIRST.
C***** VARIABLES WILL BE WRITTEN ON TAPE(FOR THE CORRELATION ROUTINE) IN
C***** THE SAME ORDER.
       READ 8,(TYPE(I),IVAL(I),I=1,NVAR)
     8 FORMAT(16(A1,I3,X))
C*****
C***** LOOK AT EACH VARIABLE READ IN.  STORE NUMBER OF COMPUTED VARIABLES
C***** READ IN LOCATION NCOMP.  STORE COUNT OF HISTORY VARIABLES IN NHIST
C***** PRINT ERROR MESSAGE IF HISTORY VARIABLE IS LARGER THAN ONE WORD.
C***** STORE COMPUTED VARIABLE NUMBERS IN ARRAY ICOMP.  STORE HISTORY
C***** VARIABLE NUMBERS IN ARRAY IVNUM.  STORE RELATIVE LOCATION OF
C***** HISTORY VARIABLES IN LHIST.  MAKE SURE THAT VARIABLES ARE
C***** SPECIFIED CORRECTLY AND IN RANGE.  CLEAR IBAD TABLE.
       DO 110 I=1,NVAR
       IBAD(I)=0
       IF(IVAL(I).EQ.0) GO TO 320
       IF(TYPE(I).EQ.V) GO TO 105
       IF(TYPE(I).NE.C) GO TO 350
       NCOMP=NCOMP+1
       ICOMP(NCOMP)=IVAL(I)
       GO TO 110
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
  105 NHIST=NHIST+1
      IF(IVAL(I).GT.K) GO TO 320
      IVNUM(NHIST)=IVAL(I)
      IK=IVAL(I)
      ITEST=IREL( IK )/ISHF18
      IF(ITEST.GT.1) GO TO 305
      LHIST(NHIST)=MOD(IREL( IK ),ISHF18)
  110 CONTINUE
      PRINT 9
    9 FORMAT('0')
      PRINT 10
   10 FORMAT('1',47HVAR NO      COUAP VARIABLE DEFINITION AND FORMAT)
C****
C**** SORT ARRAYS ICOMP AND LHIST ASCENDING SO THAT VARIABLES WILL BE
C**** EXTRACTED AND WRITTEN ON TAPE IN ASCENDING VARIABLE ORDER,
C**** COMPUTED VARIABLES FIRST.
      IF(NCOMP.LT.2) GO TO 113
      NUM=NCOMP-1
  111 IMORE=0
      DO 112 N=1,NUM
      IF(ICOMP(N).LE.ICOMP(N+1)) GO TO 112
      IHOLD=ICOMP(N)
      ICOMP(N)=ICOMP(N+1)
      ICOMP(N+1)=IHOLD
      IMORE=1
  112 CONTINUE
      IF(IMORE.NE.0) GO TO 111
  113 IF(NHIST.LT.2) GO TO 116
      NUM=NHIST-1
  114 IMORE=0
      DO 115 N=1,NUM
      IF(LHIST(N).LE.LHIST(N+1)) GO TO 115
      IHOLD=LHIST(N)
      LHIST(N)=LHIST(N+1)
      LHIST(N+1)=IHOLD
      IHOLD=IVNUM(N)
      IVNUM(N)=IVNUM(N+1)
      IVNUM(N+1)=IHOLD
      IMORE=1
  115 CONTINUE
      IF(IMORE.NE.0) GO TO 114
C****
C**** READ DICTIONARY CONTAINING VARIABLE DEFINITIONS.  PRINT THE
C**** DEFINITION FOR EACH VARIABLE SELECTED.  STORE THE SIZE(CHARS) OF
C**** EACH VARIABLE IN ARRAY IFMT.
  116 ISEQ=1
      DO 160 N=1,NDICT
       READ(IHT) K,((KC(I,J),I=1,15),J=1,K)
       ITYP=KC(1,1) / ISHF30
       IF(ITYP.NE.ICEE) GO TO 135
```

LINE #   TEXT...

```
      IF(NCOMP.EQ.0) GO TO 160
      DO 130 IX=1,K
      DO 120 KK=1,NCOMP
      IK=NC+IX
      IF(ICOMP(KK).NE.IK) GO TO 120
      PRINT 12,ISEQ,(KC(I,IX),I=1,15)
   12 FORMAT(' ',3X,I3,5X,15A6)
      ISEQ=ISEQ+1
      ILINE=ILINE+1
      IF(ILINE.NE.55) GO TO 130
      ILINE=0
      PRINT 3
      GO TO 130
  120 CONTINUE
  130 CONTINUE
      NC=NC+K
      GO TO 160
  135 IF(NHIST.EQ.0) GO TO 160
      DO 150 IX=1,K
      DO 140 KK=1,NHIST
      IK=NV+IX
      IF(IVNUM(KK).NE.IK) GO TO 140
      PRINT 12,ISEQ,(KC(I,IX),I=1,15)
      ISEQ=ISEQ+1
C****
C**** VARIABLE FORMATS MAY BE RIGHT ADJUSTED IN WORD 15 OR THEY MAY
C**** CONTAIN A TRAILING BLANK DEPENDING ON WHEN THE TAPE WAS CREATED.
      DECODE(405,KC(15,IX)) IFMT(KK)
405   FORMAT(5X,A1)
      CALL BCDBIN(IFMT(KK),IFMT(KK),1,$137)
      GO TO 138
137   DECODE(406,KC(15,IX)) IFMT(KK)
406   FORMAT(4X,A1)
      CALL BCDBIN(IFMT(KK),IFMT(KK),1,$305)
138   IF(IFMT(KK).LT.1.OR.IFMT(KK).GT.6) GO TO 305
      ILINE = ILINE + 1
      IF(ILINE.NE.55) GO TO 150
      ILINE=0
      PRINT 3
      GO TO 150
  140 CONTINUE
  150 CONTINUE
      NV=NV+K
  160 CONTINUE
C****
C**** READ CASE DATA.  EXTRACT COMPUTED VARIABLES AND BACKGROUND
C**** VARIABLES AND WRITE ON UNIT 3.  SKIP ANY CASE WHICH CONTAINS
C**** OUT OF RANGE VARIABLES.  SUBROUTINE ICVNI CONVERTS HISTORY
C**** VARIABLES FROM BCD TO BINARY.  VARIABLE DATA IS WRITTEN ON UNIT
C**** 3 SO THAT NCASE CAN BE ESTABLISHED BEFORE STARTING CORRELATION.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
C***** A COMPUTED VARIABLE WHICH WAS SET OUT OF RANGE
C***** BY PREGEN WILL EQUAL -999999.
 180    READ(IHT) K,(AA(I),I=1,K),L,(KB(I),I=1,L)
        IF(AA(1).EQ. END     ) GO TO 240
        GO TO (190,185), NJUMP
  185 IF(IXVAR.GT.K) GO TO 360
        IF(AA(IXVAR).EQ.FGOOD) GO TO 190
        LCASE = LCASE-1
        GO TO 180
  190 ICHK=0
        IF(NCOMP.EQ.0) GO TO 210
        DO 200 I=1,NCOMP
        J=ICOMP(I)
        IF(J.GT.K) GO TO 340
        ARRAY(I)=AA(J)
        IF(ARRAY(I).NE.(-99999999.)) GO TO 200
        IBAD(I)=IBAD(I)+1
        ICHK=1
  200 CONTINUE
  210 IF(NHIST.EQ.0) GO TO 230
        DO 220 K=1,NHIST
        J=LHIST(K)
        I=NCOMP+K
         CALL BCDBIN(KB(J),KB(J),IFMT(K),$215)
        ARRAY(I)=KB(J)
        GO TO 220
  215 IBAD(I)=IBAD(I)+1
        ICHK=1
  220 CONTINUE
  230 IF(ICHK.EQ.1) GO TO 180
        NCASE=NCASE+1
        JCASE=NCASE+1
         WRITE(ISCR) JCASE,(ARRAY(I),I=1,NVAR)
        GO TO 180
  240 KCASE=LCASE-NCASE
        PRINT 9
        PRINT 11,NCASE,KCASE
   11 FORMAT('KCASES USED =',I5,'.  CASES REJECTED ON INVALID VARIABLES
     1=',I4,'.')
        GO TO (245,242), NJUMP
  242 PRINT 15, IXVAR,IGOOD
   15 FORMAT(' CASES WERE USED WHEN COMPUTED VARIABLE',I3,' WAS EQUAL TO
     1 ',I2,'.')
  245  REWIND ISCR
       REWIND IHT
C*****
C***** PRINT A MATRIX SHOWING THE NUMBER OF INVALID VARIABLES FOUND.
       PRINT 30
   30 FORMAT('1VARIABLE ERROR COUNTS',/////)
       PRINT 40, (I,IBAD(I),I=1,NVAR)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
   40 FORMAT (7(1X,I3,'.',I4,6X),/)
      IF(NCASE.EQ.0) GO TO 310
C****
C**** SET UP PERSUB TYPE TAPE HEADER AND WRITE ON UNIT 2.  WRITE NCASE
C**** RECORDS AND CLOSE TAPE WITH PERSUB TYPE END RECORD.
      MINUS1=-1
      IONE=1
      ITWO=2
       WRITE(ISCR1) MINUS1,IONE,NCASE,NVAR,ITWO,ID(1),ID(2)
      DO 250 IX=1,NCASE
       READ(ISCR) JCASE,(ARRAY(I),I=1,NVAR)
       WRITE(ISCR1) JCASE,(ARRAY(I),I=1,NVAR)
  250 CONTINUE
      I=999999999
      J=NCASE+2
       WRITE(ISCR1) I,J
       REWIND ISCR1
C****
C**** CALL CORRELATION ROUTINE
       CALL CORRLB(ISCR1,ID(1),NCASE,NVAR)
C****
C**** SAVE CORRELATION MATRIX ON UNIT 3.
       CALL TAPGEN(IOUT)
      NC=NVAR*(NVAR+2)+2
      K=NC-1
      A(K)=NCASE
      A(K+1)=NVAR
       CALL CRTOTP(1,1,NC,1,IOUT,12,ID(1))
C****
C**** READ A CARD.  SHOULD BE 'REGR' OR 'END'.
       READ(5,13,END=401) ANAME,ITER,ID(1),ID(2)
   13 FORMAT(A6,I1,X,A6,A2)
      GO TO 295
C****
C**** READ CORR MATRIX INTO CORE FOR REGRESSION PROBS.
  290  KCORR = 10
      ITER=NVAR
       READ(KCORR) SKIP,SKIP,NR,NC
       REWIND KCORR
       CALL TPTOCR(KCORR,ID(1),NR,NC,1,0)
      K=NC-1
      NCASE=A(K)
      NVAR=A(K+1)
C****
C**** CALL REGRESSION ROUTINE
  295 CALL REGREF(NVAR,NCASE,ITER)
      GO TO 401
C****
C**** ERROR MESSAGES
  300 PRINT 14, JA(1),JA(2)
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
   14 FORMAT('1STUDY ID FROM HISTORY OR KPATH FILE DOES NOT MATCH CONTRO
     1L CARD.  FILE ID IS ',A6,A2)
      GO TO 400
  305 PRINT 16, IK
   16 FORMAT(18H1HISTORY VARIABLE ,I3,95H IS NOT BETWEEN 1 AND 6 CHARACT
     1ERS ACCORDING TO THE VARIABLE DICTIONARY ENTRY.  RUN TERMINATED.)
      GO TO 400
  310 PRINT 18
   18 FORMAT( 99H1ALL CASE DATA REJECTED.  A BACKGROUND VARIABLE CONTAIN
     1ING ONLY ALPHA DATA MAY HAVE BEEN REQUESTED.)
      GO TO 400
  320 PRINT 20, IVAL(I)
   20 FORMAT(21H1BACKGROUND VARIABLE ,I3,34H DOES NOT EXIST ON THE CODAP
     1 TAPE.)
      GO TO 400
  330 PRINT 22
   22 FORMAT( 80H1THE NUMBER OF VARIABLES SPECIFIED ON THE VARIABLE REQU
     1EST CARD IS OUT OF RANGE.)
      GO TO 400
  340 PRINT 24, J
   24 FORMAT(18H1COMPUTED VARIABLE,I3,34H DOES NOT EXIST ON THE CODAP TA
     1PE.)
      GO TO 400
  350 PRINT 26, IVAL(I)
   26 FORMAT(17H1VARIABLE NUMBER ,I3,59H FROM THE VARIABLE REQUEST CARD
     1DOES NOT BEGIN WITH V OR C.)
      GO TO 400
  360 PRINT 24, IXVAR
400   CALL ERTRAN(2)
401   WRITE(6,370)
  370  FORMAT('1')
       STOP
       END
```

ALPHA   LOADER OF ALPHA CHARS SUBROUTINE (FROM PERSUB)

CORRES PAGE 69

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C      ALPHA                 27 SEPTEMBER 1963
C      GENERATES ALPHA INFORMATION
       SUBROUTINE ALPHA(ALPH,NUMHOL,HOLLER)
       DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
       COMMON FMT1,FMT2,A
       EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
       DIMENSION ALPH(1),HOLLER(1)
    10 M=(NUMHOL +5)/6
    20 DO 30 I=1,M
    30 ALPH(I)=HOLLER(I)
    40 RETURN
    50 END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C      CORRLB              1 AUG 1967
C      MEANS,STANDARD DEVIATIONS,CORRELATIONS
C      USES SIGMA AS BUFFER FROM TAPE
       SUBROUTINE CORRLB(IDTAPE,FILE,NUM,NVAR)
       DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
       COMMON FMT1,FMT2,A
       EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
       LMEAN=1
       LSIGMA=1+NVAR
       LCORR=1+2*NVAR
    20 CALL ZEROST(LCORR,NVAR,NVAR)
    21 CALL ZEROST(LMEAN,NVAR,1)
    22 CALL ZEROST(LSIGMA,NVAR,1)
    23 CALL POSTAP(LOCREC,IDTAPE,0,FILE)
    40 ITO=LSIGMA+NVAR-1
    30 DO  74  I=1,NUM
    41 READ(IDTAPE)LOCT,(A(II),II=LSIGMA,ITO)
       IM=LMEAN-1
       IA=LSIGMA-1
       IT=LCORR-1
    51 DO  74  J=1,NVAR
       IM=IM+1
       IA=IA+1
       IT=IT+J-1
       IB=LSIGMA+J-2
       A(IM)=A(IM)+A(IA)
    70 DO  74  L=J,NVAR
       IT=IT+1
       IB=IB+1
    74 A(IT)=A(IT)+A(IA)*A(IB)
    75 REWIND IDTAPE
C      COMPUTE R MATRIX
    78 CALL ZEROST(LSIGMA,NVAR,1)
    80 FN=NUM
C      COMPUTE NON DIAGONAL ELEMENTS OF R MATRIX
       KM1=NVAR-1
    81 DO 130 I=1,KM1
    82 IP1=I+1
    83 DO 130 J=IP1,NVAR
    84 ISI=LCORR-1+(I-1)*NVAR+I
    85 ISJ=LCORR-1+(J-1)*NVAR+J
    86 ISJI=LCORR-1+(J-1)*NVAR+I
    87 ISIJ=LCORR-1+(I-1)*NVAR+J
    88 IMI=LMEAN-1+I
    89 IMJ=LMEAN-1+J
    90 DEN=SQRT(ABS((FN*A(ISI)-A(IMI)*A(IMI))*(FN*A(ISJ)-A(IMJ)*A(IMJ))))
    95 IF(DEN) 110,100,110
   100 A(ISIJ)=0.0
   105 GO TO 130
   110 A(ISIJ)=(FN*A(ISIJ)-A(IMI)*A(IMJ))/DEN
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
  115 IF(ABS(A(ISIJ))-1.0) 130,130,120
  120 PRINT 125,I,J,A(ISIJ),A(ISIJ)
  125 FORMAT(' OUT OF RANGE CORRELATION I=',I5,4X,'J=',I5,4X,'R=',
     1 F10.4,'   IN OCTAL=',O12,' R IS SET = 1.000000')
      A(ISIJ) = 1.0000
  126 CONTINUE
  130 A(ISJI)=A(ISIJ)
C      COMPUTE MEAN AND SIGMA
  135 DO 165 I=1,NVAR
  140 IM=LMEAN-1+I
  145 A(IM)=A(IM)/FN
  150 IS=LSIGMA-1+I
  155 II=LCORR-1+(I-1)*NVAR+I
  160 A(IS)=SQRT(ABS((A(II)/FN)-A(IM)*A(IM)))
C      COMPUTE DIAGONAL ELEMENTS OF R MATRIX
  161 IF(A(IS))162,162,164
  162 A(II)=0.0
  163 GO TO 165
  164 A(II)=1.0
  165 CONTINUE
  167 CALL PRIMSC(LMEAN,LSIGMA,LCORR,NVAR)
  170 RETURN
      END
```

CRTOTP   CORE TO TAPE SUBROUTINE (FROM PERSUB)

COMREG PAGE   72

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C     CRTOTP               21 JANUARY 1969
C     CORE TO TAPE
      SUBROUTINE CRTOTP(LFRM,NR,NC,IDFMT,IDTAPE,NUMMOL,HOLLER)
      DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT[illegible],A(1),KA(1)
      COMMON FMT1,FMT2,A
      EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
      DIMENSION HOLLER(22)
    1 IF(NUMMOL-12)2,10,10
    2 PRINT 3, NUMMOL
    3 FORMAT(51H ERROR IN CRTOTP.  NUMMOL IS LESS THAN 12. NUMMOL =,I4)
    4 CALL EXIT
   10 CALL POSTAP (LOCREC,IDTAPE,999999999,HOLLER)
   15 KOUNT=-LOCREC
   20 NWORDS=(NUMMOL+5)/6
   30 WRITE(IDTAPE)KOUNT,IDFMT,NR,NC,NWORDS,(HOLLER(I),I=1,NWORDS)
   31 IF(IDFMT) 40, 73,40
   40 DO  70   I=1,NR
   41 KOUNT=LOCREC+I
   50 KFR=LFRM+(I-1)*NC
   60 KTO=KFR-1+NC
   70 WRITE(IDTAPE)KOUNT,(A(J),J=KFR,KTO)
   71 GO TO 80
   72 FORMAT(I9,13A6,A2)
   73 DO 77  I=1,NR
   74 KOUNT= LOCREC+I
   75 KFR=LFRM+(I-1)*14
   76 KTO=KFR+13
   77 WRITE(IDTAPE, 72 )KOUNT,(A(J),J=KFR,KTO)
   80 KOUNT=LOCREC+NR+1
   90 I=999999999
  100 WRITE(IDTAPE) I,KOUNT
  110 REWIND IDTAPE
  120 RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C     PLEVEL              31 OCTOBER 1966
      SUBROUTINE PLEVEL(DF1,DF2,F,P)
      DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
      DIMENSION Y(6),ARG(3),GAMA(3)
      COMMON FMT1,FMT2,A
      EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
      BASMLN(Z,X) = (X+.5)*ALOG(X)-X+.918938534+(.833333333E-1-Z*(.27777
     17778E-2-Z*(.793650794E-3-Z*(.595238095E-3-Z*.841750842E-3))))/X
      IF(DF1.LT.1..OR.DF2.LT.1..OR.F.LT.0.)GO TO 14
      IF(ARG(1).EQ.DF1.AND.ARG(2).EQ.DF2)GO TO 7
      ARG(1) = DF1
      ARG(2) = DF2
      ARG(3) = DF1+DF2
      DO 6 I=1,3
      IF(ARG(I).EQ.1.)GO TO 5
      T = ARG(I)-2.
      J = AMOD(ARG(I),2.)+1.
      GO TO (2,1),J
    1 U = (T-1.)*.5
      GAMA(I) = .572364943-T*.693147181
      IF(U.LT.2.)GO TO 25
      GO TO 3
    2 T = T*.5
      GAMA(I) = 0.
   25 IF(T.LT.2.)GO TO 6
      GO TO 4
    3 Z = 1./(U*U)
      GAMA(I) = GAMA(I)-BASMLN(Z,U)
    4 Z = 1./(T*T)
      GAMA(I) = GAMA(I)+BASMLN(Z,T)
      GO TO 6
    5 GAMA(I) = .572364943
    6 CONTINUE
      C = GAMA(1)+GAMA(2)-GAMA(3)-69.-.693147181
      Y(1) = .4054651181
      Y(2) = -1.203972804
      Y(3) = .5877866649
      Y(4) = Y(2)
      Y(5) = Y(1)
    7 AX = DF2/(F*DF1+DF2)
      IF (AX.GT.0.99999900) GO TO 136
      H = ATAN(SQRT(AX/(1.-AX)))/60.
      IF(H.LE..130899694E-1)GO TO 8
      H = .2617993387E-1-H
      CN = DF1-1.
      CM = (DF2-1.)*.5
      P = -1./H
      XM = -H
      GO TO 9
    8 CN = DF2-1.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      CM = (DF1-1.)*.5
      P = 0.
      XM = M
    9 IF (CN.NE.0.) GO TO 95
      XX = Y(2)-C
      IF(XX.LT.0.)GO TO 95
      P=P+EXP(XX-69.)
   95 Y(6) = -.5108256238
      X = 0.
      DO 13 I=1,10
      GO TO(11,11,11,11,11,11,11,11,11,10),I
   10 Y(6) = Y(2)
   11 DO 13 J=1,6
      X = X+H
      XS = SIN(X)
      Z = Y(J)-C+CN*ALOG(XS)+CM*ALOG(1.-XS*XS)
      IF(Z)13,12,12
   12 P = P+EXP(Z-69.)
   13 CONTINUE
      P = XM*P
      IF (P.LT.1.0E-05 ) P = 0.
  135 RETURN
  136 P = 1.
      GO TO 135
   14 P = 1.
      PRINT15,DF1,DF2,F
   15 FORMAT(7H1DF1 = F5.0,7H,DF2 = F5.0,5H,F = F10.5//
     148H P DOES NOT EXIST IF DF1 OR DF2 IS LESS THAN 1   /
     248H OR IF F IS LESS THAN ZERO.                        /
     348H P FOR THIS PROBLEM HAS BEEN ABRITRARILY SET      /
     448H EQUAL TO 1. AND A NORMAL RETURN HAS OCCURED.     )
      GO TO 135
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C      POSTAP                  8 FEBRUARY 1965
C      POSITIONS TAPE READY TO READ OR WRITE RECORD
       SUBROUTINE POSTAP (LOCREC,IDTAPE,IDREC,KFILE)
       DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
       COMMON FMT1,FMT2,A
       EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
       DIMENSION KFILE(2)
    10 REWIND IDTAPE
    11 IRECRD=1
    12 FORMAT(I9,13A6,A2)
    13 LAST=0
    20 IF(IRECRD) 21, 23,21
    21 READ (IDTAPE)I
    22 GO TO 30
    23 READ (IDTAPE,12)I
    24 IF(I-LAST)30,25,30
    25 IRECRD=1
    30 IF(I)50 ,110,90
    50 BACKSPACE IDTAPE
    60 READ(IDTAPE) I,ITYPE,NR,NC,NWORDS,KNAME1,KNAME2
    70 IF(IDREC) 110, 71,104
    71 IF(KFILE(1)-KNAME1) 75, 72, 75
    72 IF(KFILE(2)-KNAME2) 75, 73, 75
    73 LOCREC=-I+1
    74 GO TO 106
    75 IF(ITYPE) 20, 76,20
    76 LAST=-I+NR
    77 IRECRD=0
    78 GO TO 20
    90 IF (I-999999999)103,100,110
   100 IF (IDREC-999999999)110,101,110
   101  BACKSPACE IDTAPE
   102  READ(IDTAPE) I,J
       I=J
        GO TO 105
   103 IF(IDREC-I) 20,105,20
   104 IF(IDREC-IABS(I)) 75,105, 75
   105 LOCREC=I
       BACKSPACE IDTAPE
   106 RETURN
  110  PRINT 111, KNAME1,KNAME2,KFILE(1),KFILE(2)
  111  FORMAT('1CORRELATION MATRIX ID NO MATCH.  TAPE IS ',A6,A2,
      1'.  CARD IS ',A6,A2,'.')
   112 CALL EXIT
       END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C      PRIMSC                 31 OCTOBER 1966
C      PRINTS MEANS,STANDARD DEVIATIONS, CORRELATIONS
       SUBROUTINE PRIMSC(LMEAN,LSIGMA,LCORR,NVAR)
       DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
       COMMON FMT1,FMT2,A
       EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
       LM=LMEAN-1
       LS=LSIGMA-1
       DO 5 I=1,NVAR
       I2 = I-1
       IF (MOD(I2,50).NE.0) GO TO 4
       PRINT 1
     1 FORMAT (1H1,3X,6HVAR NO,8X,4HMEAN,8X,4HS.D.)
     4 LM=LM+1
       LS=LS+1
     5 PRINT 6, I,A(LM),A(LS)
     6 FORMAT (I10,2F12.4)
       CALL PRINTI(LCORR,NVAR,NVAR,1,17,17H      CORRELATIONS)
   121 RETURN
   122 END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C       PRINT                    31 OCT 1966
C       PRINTS MATRIX PARTIONED BY COLUMNS USING FORMAT IN FMT2
        SUBROUTINE PRINT1(LARRAY,NR,NC,IDFMT,NUMMOL,HOLLER)
        DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
        COMMON FMT1,FMT2,A
        EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
        DIMENSION HOLLER(22)
     30 IF (IDFMT)35,60,35
     35 GO TO (40,51,55,59),IDFMT
     40 CALL ALPHA (FMT2,15,15H(1H 19,10F12.4))
     50 GO TO 74
     51 CALL ALPHA (FMT2,13,13H(1H 19,10I12))
     52 GO TO 74
     55 CALL ALPHA (FMT2,12,12H(1H 19,20A6))
     56 KSIZE=20
     57 CALL ALPHA(FMT1,19,19H( //8H/        20I6))
     58 GO TO 76
     59 CALL ALPHA (FMT1,18,18H( //6H/      ,6I20))
    500 CALL ALPHA (FMT2,14,14H(1H 19,6E20.8))
    501 KSIZE=6
    502 GO TO 76
     60 READ 61,(FMT2(I),I=1,15)
     61 FORMAT (13A6,A1,F1.0)
     62 IF (FMT2(15)) 63,74,63
     63 READ 61,(FMT2(I),I=15,22)
     74 KSIZE=10
     75 CALL ALPHA(FMT1,20,20H( //8H/        10I12))
     76 KMANY=(((NC -1)/KSIZE)+1)*KSIZE
        IF (NR.GT.25) GO TO 79
        NBLOCK=60/(NR+4)
        IBLOCK=0
        GO TO 80
     79 NBLOCK=1
        IBLOCK=0
     80 DO 210  L=KSIZE,KMANY,KSIZE
     90 L1=L-(KSIZE-1)
    100 IF(L-KMANY)130,110,130
    110 L2=NC
    120 GO TO 140
    130 L2=L
    140 KSTAR=LARRAY+L-KSIZE-NC
        KSTOP=KSTAR+KSIZE-1
        DO 210 I=1,NR
        I2=I-1
        IF (MOD(I2,50).NE.0) GO TO 160
        IF (IBLOCK.EQ.0) GO TO 8
        IF (IBLOCK .LT. NBLOCK) GO TO 11
      8 IBLOCK=1
        GO TO 9
     11 IBLOCK=IBLOCK+1
```

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      GO TO 147
    9 IF (NUMHOL) 145,145,10
   10 N=(NUMHOL+5)/6
   20 PRINT 21,(HOLLER(K),K=1,N)
   21 FORMAT (1H1,21A6,A5///)
      GO TO 147
  145 PRINT 146
  146 FORMAT (1H1)
  147 PRINT FMT1,(K,K=L1,L2)
  160 KSTAR=KSTAR+NC
  170 IF(L-KMANY)180,200,180
  180 KSTOP=KSTOP+NC
  190 GO TO 210
  200 KSTOP=LARRAY-1+1*NC
  210 PRINT FMT2,I,(A(J),J=KSTAR,KSTOP)
  220 RETURN
  230 END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C       REGREF                  1 AUG 1967
        SUBROUTINE REGREF(LOCM,LOCSD,LOCCOR)
        DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
        DIMENSION PROBID(2),MFLD(125),KV(125), IENT(125)
        COMMON FMT1,FMT2,A
        EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA),(MFLD,IENT)
        EQUIVALENCE (WBLANK,KBLANK)
        DOUBLE PRECISION RSQNEW,RSQOLD,ZSQRT,DTERM,BIRW,D22,A25,A32,A23
        DOUBLE PRECISION D33,A35,A42,A24,A43,A34,D44,W4,W3,W2,W1,RSQ
        DOUBLE PRECISION DSQRT,DBLE,W1BST,W2BST,W3BST,W4BST,RBST
C
C       DOUBLE PRECISION IS USED IN THIS SUBROUTINE.  REPROGRAMING TO
C       SINGLE PRECISION IS NOT ADVISED.  ACCURACY OF FINAL RESULTS DEPEND
C       UPON MAINTAINING AN ACCURATE 7 TO 8 SIGNIFICANT DIGITS (DECIMAL
C       DIGITS) IN INTERMEDIATE COMPUTATIONS.
C
        SUMS(ALSD,R1,R2)=CASES*ALSD*ALSD*(R1-R2)
        KBLANK=-17997958192
        KFTST=1
        NVAR=LOCM
        CASES=LOCSD
   1000 IPRT=LOCCOR
   1001 LMEAN=1
        LSIGMA=1+NVAR
        LCORR=1+2*NVAR
        LSTDWT=1+(2+NVAR)*NVAR
        LWTS=1+(3+NVAR)*NVAR
        LRSQ=2+(4+NVAR)*NVAR
   1004 KRSQ=LRSQ-1
C       INITIALIZE FOR PROBLEM
        NUM=19
    500 READ(5,1,END=503)PROBID,STOP,ITKNT,NFLDS,(MFLD(I),I=1,NUM)
      1 FORMAT(2A5,F8.8,I2,20I3)
      2 IF(NFLDS.LE.NUM) GO TO 5
        JJ=NUM+1
        NUM=NUM+19
        READ 26,(MFLD(I),I=JJ,NUM)
     26 FORMAT(23X,19I3)
        GO TO 2
      5 IF (NFLDS.EQ.8) GO TO 501
        KFTST=1
        DO 890 I=1,NFLDS
        IF(MFLD(I).GT.NVAR.OR.MFLD(I).LE.0) GO TO 891
    890 CONTINUE
        GO TO 6
    891 PRINT 892, PROBID
    892 FORMAT( 67H1REGRESSION CONTROL CARD CONTAINS AN INVALID VARIABLE.
      1 PROB. ID = ,2A5)
        GO TO 503
      6 NC=MFLD(NFLDS)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      8 NFLD1=NFLDS-1
      3 PRINT 4,PROBID,STOP,ITKNT,NC,(MFLD(I),I=1,NFLD1)
      4 FORMAT(24H1REGRESSION CONTROL CARD/18H PROBLEM IDENT ** 2A5,15X,15
       1HSTOP CRITERIA =F11.8,5X,25HITERATION LIMIT CONTROL =I3/10H CRITER
       2ION14,10X,10HPREDICTORS,6(9X,I3,1H-,I3),(/1H ,33X,6(9X,I3,1H-,I3))
       3)
        LN=NC+LSIGMA-1
      7 LNC=LWTS+NC-1
        KRSQ=KRSQ+1
        K=1
        KPSW=1
        KNTP=0
        DO 750 I=2,NFLD1,2
        I1=MFLD(I-1)
        I2=MFLD(I)
        KNTP=KNTP+I2-I1+1
        DO 750 J=I1,I2
        KV(K)=J
    750 K=K+1
        IENT(1)=0
        CALL TMTCSC(124,IENT(2),IENT(1))
        KNTP1=KNTP-1
        IF (KNTP.LE.3) ITKNT=1
        IF (KNTP-2) 751,752,9
    751 KPSW=KPSW+1
    752 KPSW=KPSW+1
      9 IF (ITKNT) 503,10,11
     10 ITKNT=99999
     11 A(LSTDWT)=0.
     12 CALL TMTCSC(NVAR-1,A(LSTDWT+1),A(LSTDWT))
     13 A(LWTS)=0.
     14 CALL TMTCSC(NVAR,A(LWTS+1),A(LWTS))
     15 ZSQRT=0.D0
     16 NCN=NVAR*(NC-1)+LCORR-1
     17 ITNUM=0
     18 KSWCH=1
     19 RSQOLD=0.D0
        IF (IPRT.NE.1) GO TO 22
     20 PRINT 21
     21 FORMAT(//96H    IT       I   CORRECTION       J   CORRECTION        K
       1  CORRECTION     MULTIPLIER                    RSQ,10X,8HERR S.S.//)
C       INITIALIZE FOR ITERATION
     22 DTERM=0.D0
        RSQNEW=0.D0
        W1BST=0.D0
        W2BST=0.D0
        W3BST=0.D0
        W4BST=0.D0
        B1RW=0.D0
        GO TO (23,23,753),KPSW
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
  753 IBEST=KV(1)
      JBEST=IBEST
      KBEST=JBEST
      NCI=NCN+IBEST
      LI=IBEST+LWTS-1
      W2BST=(DBLE(A(NCI))-DBLE(A(LI))*DBLE(A(LNC)))/(1.D0-DBLE(A(LI))*DB
     1LE(A(LI)))
      W1BST=DBLE(A(LNC))-DBLE(A(LI))*W2BST
      RSQNEW=DBLE(A(LNC))*W1BST+DBLE(A(NCI))*W2BST
      RSQOLD=RSQNEW
      GO TO 82
C     SEARCH FOR BEST PAIR
C     SOLVE WTS FOR COMPOSITE AND THREE SELECTED VARIABLES
   23 DO 55 IP=1,KNTP1
      I=KV(IP)
      IP1=IP+1
   27 NCI=NCN+I
   28 LI=LWTS+I-1
   29 IR=NVAR*(I-1)+LCORR-1
   30 D22=1.D0-DBLE(A(LI))*DBLE(A(LI))
   31 A24=(DBLE(A(NCI))-DBLE(A(LI))*DBLE(A(LNC)))/D22
   32 DO 55 JP=IP1,KNTP
      J=KV(JP)
   37 IJ=IR+J
   38 IF (ABS(A(IJ))-1.0) 39,55,55
   39 LJ=LWTS+J-1
   40 NCJ=NCN+J
   41 A32=DBLE(A(IJ))-DBLE(A(LI))*DBLE(A(LJ))
   42 A23=A32/D22
   43 D33=1.D0-DBLE(A(LJ))*DBLE(A(LJ))-A32*A23
   44 DTERM=D22*D33
   45 IF (DTERM.LT.1.D-08.OR.DTERM.GT.1.D0) GO TO 55
   46 W3=(DBLE(A(NCJ))-DBLE(A(LJ))*DBLE(A(LNC))-A32*A24)/D33
   47 W2=A24-A23*W3
   48 W1=DBLE(A(LNC))-DBLE(A(LI))*W2-DBLE(A(LJ))*W3
   49 RSQ=W1*DBLE(A(LNC))+W2*DBLE(A(NCI))+W3*DBLE(A(NCJ))
   50 DIFF=RSQ-RSQNEW
   51 IF (DIFF) 55,55,52
   52 RSQNEW=RSQ
   53 IBEST=I
   54 JBEST=J
   55 CONTINUE
   56 IF (RSQNEW) 754,754,58
  754 ITKNT=1
      KPSW=3
      GO TO 22
   58 LI=LWTS+IBEST-1
      IR=NVAR*(IBEST-1)+LCORR-1
      JR=NVAR*(JBEST-1)+LCORR-1
   59 LJ=LWTS+JBEST-1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
   60 NCI=NCN+IBEST
   61 NCJ=NCN+JBEST
      D22=1.D0-DBLE(A(LI))*DBLE(A(LI))
      A25=(DBLE(A(NCI))-DBLE(A(LI))*DBLE(A(LNC)))/D22
      IJ=IR+JBEST
      A32=DBLE(A(IJ))-DBLE(A(LI))*DBLE(A(LJ))
      A23=A32/D22
      D33=1.D0-DBLE(A(LJ))*DBLE(A(LJ))-A32*A23
   77 A35=(DBLE(A(NCJ))-DBLE(A(LJ))*DBLE(A(LNC))-A32*A25)/D33
      KBEST=0
      W3BST=A35
      W2BST=A25-A23*W3BST
      W1BST=DBLE(A(LNC))-DBLE(A(LI))*W2BST-DBLE(A(LJ))*W3BST
      RBST=RSQNEW
      GO TO (63,82,221),KPSW
   63 DO 80 KP=1,KNTP
      K=KV(KP)
   67 IF (K.EQ.IBEST.OR.K.EQ.JBEST) GO TO 80
   68 IK=IR+K
   69 JK=JR+K
   70 IF (ABS(A(IK)).EQ.1.0.OR.ABS(A(JK)).EQ.1.0) GO TO 80
   71 LK=LWTS+K-1
      NCK=NCN+K
   72 A42=DBLE(A(IK))-DBLE(A(LI))*DBLE(A(LK))
   73 A24=A42/D22
   74 A43=DBLE(A(JK))-DBLE(A(LJ))*DBLE(A(LK))-A42*A23
   75 A34=A43/D33
   76 D44=1.D0-DBLE(A(LK))*DBLE(A(LK))-A42*A24-A43*A34
      DTERM=D22*D33*D44
      IF (DTERM.LT.1.D-08.OR.DTERM.GT.1.D0) GO TO 80
      W4=(DBLE(A(NCK))-DBLE(A(LK))*DBLE(A(LNC))-A42*A25-A43*A35)/D44
  757 W3=A35-A34*W4
      W2=A25-A24*W4-A23*W3
      W1=DBLE(A(LNC))-DBLE(A(LK))*W4-DBLE(A(LJ))*W3-DBLE(A(LI))*W2
   78 RSQ=DBLE(A(LNC))*W1+DBLE(A(NCI))*W2+DBLE(A(NCJ))*W3+DBLE(A(NCK))*W
     14
      DIFF=RSQ-RSQNEW
      IF (DIFF) 80,80,79
   79 RSQNEW=RSQ
      KBEST=K
      W1BST=W1
      W2BST=W2
      W3BST=W3
      W4BST=W4
   80 CONTINUE
   82 ITNUM=ITNUM+1
      IF (KBEST) 760,763,63
  760 IF (ITNUM.EQ.0) ITKNT=1
      RSQNEW=RBST
  759 KBEST=JBEST
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      KPSW=2
   83 DIFF=RSQNEW-RSQOLD
   84 IF (DIFF.LE.STOP) KSWCH=2
   85 RSQOLD=RSQNEW
   86 A(LNC)=DSQRT(RSQNEW)
   87 IF (RSQOLD-1.) 91,91,88
   88 PRINT 89,RSQOLD
   89 FORMAT(//41H PROBLEM TERMINATED.  RSQ GT 1.0.  RSQ = ,F15.8)
   90 GO TO 500
   91 BIRW=W1BST/ZSQRT
   92 ZSQRT=A(LNC)
   93 KI=NVAR*(IBEST-1)+LCORR-1
   94 KJ=NVAR*(JBEST-1)+LCORR-1
   95 KK=NVAR*(KBEST-1)+LCORR-1
   96 DO 106 JP=1,KNTP
      I=KV(JP)
  100 LI=KI+1
  101 LJ=KJ+1
  102 LK=KK+1
  103 LLI=LSTDWT+I-1
  104 A(LLI)=DBLE(A(LLI))*BIRW
  105 LLI=LWTS+I-1
  106 A(LLI)=(W1BST*DBLE(A(LLI))+W2BST*DBLE(A(LI))+W3BST*DBLE(A(LJ))+W4B
     1ST*DBLE(A(LK)))/ZSQRT
  107 LI=LSTDWT+IBEST-1
  108 LJ=LI-IBEST+JBEST
  109 LK=LI-IBEST+KBEST
  110 A(LI)=DBLE(A(LI))+W2BST
  111 A(LJ)=DBLE(A(LJ))+W3BST
      A(LK)=DBLE(A(LK))+W4BST
      IF (IENT(IBEST).EQ.0) IENT(IBEST)=ITNUM
      IF (IENT(JBEST).EQ.0) IENT(JBEST)=ITNUM
      IF (IENT(KBEST).EQ.0) IENT(KBEST)=ITNUM
  112 IF (IPRT.NE.1) GO TO 114
      ESS=SUMS(A(LN),1.,RSQNEW)
      GO TO (763,762,761),KPSW
  761 W3BST=WBLANK
      JBEST=KBLANK
  762 W4BST=WBLANK
      KBEST=KBLANK
  763 GO TO (905,906,907),KPSW
  905 PRINT 805,ITNUM,IBEST,W2BST,JBEST,W3BST,KBEST,W4BST,BIRW,RSQNEW,ES
     1S
  805 FORMAT(1H ,I5,3(4X,I3,F13.8),2F15.8,5X,F13.4)
      GO TO 114
  906 PRINT 806,ITNUM,IBEST,W2BST,JBEST,W3BST,KBEST,W4BST,BIRW,RSQNEW,ES
     1S
  806 FORMAT(1H ,I5,2(4X,I3,F13.8),2A6,8X,2F15.8,5X,F13.4)
      GO TO 114
  907 PRINT 807,ITNUM,IBEST,W2BST,JBEST,W3BST,KBEST,W4BST,BIRW,RSQNEW,ES
```

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      15
  807 FORMAT(1H ,I5,4X,I3,F13.8,2(2A6,8X),2F15.8,5X,F13.4)
  114 IF (ITNUM.LT.ITKNT) GO TO 118
  115 PRINT 116,ITNUM
  116 FORMAT(/17H ITNUM = ITKNT = ,I3)
  117 KSWCH=2
  118 GO TO (22,119),KSWCH
C     COMPUTE TEST RSQ
  119 SUMT = 0.
      IF (IPRT.EQ.1) GO TO 1011
      ESS=SUMS(A(LN),1.,RSQNEW)
      PRINT 1005,ITNUM,RSQNEW,ESS
 1005 FORMAT(//18H ITERATION COUNT =I5,12X,5HRSQ =F11.8,12X,10HERR S.S.
     1=F13.4)
 1011 DO 991 KKK=1,NVAR
      IF (KKK.EQ.NC) GO TO 991
      KKL = LSTDWT+KKK-1
      KKJ = NCN+KKK
      SUMT = SUMT+A(KKL)*A(KKJ)
  991 CONTINUE
      PRINT 992,SUMT
  992 FORMAT(//28H TEST RSQ = SUM W(I)*V(I) = ,F11.8)
C     DATA PRINT
      PRINT 120
  120 FORMAT(// 78H VAR. NUM    STD. WEIGHT       RAW WEIGHT         EQN.
     1ERROR         VALIDITY,3X,5HENTRY/)
      KONST = LWTS+NVAR
  123 DO 137 JP=1,KNTP
      I=KV(JP)
  127 NCI=NCN+I
  128 LI=LWTS+I-1
  129 LJ=LSTDWT+I-1
  130 LK=LSIGMA+I-1
  131 LL=LMEAN+I-1
C     COMPUTE EQN ERROR
  132 ERR=A(LI)*ZSQRT-A(NCI)
C     COMPUTE RAW SCORE WT
  133 A(LI)=A(LJ)*A(LN)/A(LK)
  134 A(KONST) = A(KONST)+A(LI)*A(LL)
  135 PRINT 136,I,A(LJ),A(LI),ERR,A(NCI),IENT(I)
  136 FORMAT(6X,I3,F15.8,F16.8,3X,F15.8,4X,F11.8,4X,I4)
  137 CONTINUE
C     REGRESSION CONSTANT
  138 LM=LMEAN-1+NC
  139 A(KONST) = A(LM)-A(KONST)
  140 PRINT 141,A(KONST)
  141 FORMAT(///23H REGRESSION CONSTANT = ,2X,F15.8/29H (COEFFICIENT OF
     1UNIT VECTOR))
  142   A(KRSQ)=RSQOLD
  143 GO TO 500
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C       F-RATIO
  501 DF1=MFLD(3)*1000+MFLD(4)
      DF2=LOCSD-(MFLD(5)*1000+MFLD(6))
  145 K8=MFLD(1)+LRSQ-1
  146 K9=MFLD(2)+LRSQ-1
  147 F=((A(K8)-A(K9))/DF1)/((1.0-A(K8))/DF2)
  148 CALL PLEVEL(DF1,DF2,F,P)
      KDF1 = DF1
      KDF2 = DF2
      IF (KFTST.EQ.1) PRINT 149
  149 FORMAT(///18X,6HRSQFUL,11X,6HRSQRES/11H PROB IDENT,5X,2HID,5X,5HVA
     1LUE,5X,2HID,5X,22HVALUE   DF-NUM  DF-DEN,8X,1HF,11X,4HPROB)
      KFTST=0
      PRINT 150, PROBID, MFLD(1), A(K8), MFLD(2), A(K9),KDF1,KDF2,F,P
  150 FORMAT(1H0,2A5,2(3X,I4,3X,F7.4),1X,2(3X,I5),F13.4,F11.5)
      NCRIT=MFLD(7)
      NBASE=CASES
      IF (NCRIT.EQ.0.OR.NBASE.EQ.0) GO TO 500
      LN=LSIGMA+NCRIT-1
      SUM=SUMS(A(LN),A(K8),A(K9))
      PRINT 151,NCRIT,NBASE,A(LN),SUM
  151 FORMAT(87X,6HCRIT =I4,4X,3HN =I6,4X,7HSIGMA =F10.4/87X,28HSUM OF S
     1QUARES (QFUL-QRES) =F11.4)
      GO TO 500
  503 RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C      TAPGEN              26 SEPTEMBER 1963
C      GENERATES A TAPE READY FOR USE BY OTHER TAPE ROUTINES
       SUBROUTINE TAPGEN (IDTAPE)
       DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
       COMMON FMT1,FMT2,A
       EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
    10 I=999999999
    11 J=1
    20 REWIND IDTAPE
    30 WRITE (IDTAPE) I,J
    40 REWIND IDTAPE
       RETURN
       END
```

PROGRAM TYPE: SUBPROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C      TPTOCR                21 JANUARY 1964
       SUBROUTINE TPTOCR(IDTAPE,FILE,NR,NC,LTO,IDFMT)
       DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
       COMMON FMT1,FMT2,A
       EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
       DIMENSION FILE(2)
    10 CALL POSTAP (LOCREC,IDTAPE,0,FILE)
    11 BACKSPACE IDTAPE
    12 READ (IDTAPE)I,ITYPE
       IF(ITYPE)50,13,50
    13 IF(IDFMT)14,20,14
    14 GO TO( 30, 15 ),IDFMT
    15 READ 16,(FMT1(I),I=1,14)
    16 FORMAT(13A6,1X,F1.0)
    17 IF(FMT1(14)) 18, 30,18
    18 READ 16,(FMT1(I),I=14,22)
    19 GO TO 30
    20 CALL ALPHA(FMT1(1),12,12H(9X,13A6,A2))
    30 DO  33  I=1,NR
    31 KFR=LTO+(I-1)*NC
    32 KTO=KFR-1+NC
    33 READ(IDTAPE,FMT1) (A(J),J=KFR,KTO)
    34 GO TO 90
    50 DO  80    I=1,NR
    60 KFR=LTO+(I-1)*NC
    70 KTO=KFR-1+NC
    80 READ(IDTAPE)LOCREC,(A(J),J=KFR,KTO)
    90 REWIND IDTAPE
   100 RETURN
   110 END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C     ZEROST                  4 APRIL 1963
C     ZERO  PART OF STORAGE
      SUBROUTINE ZEROST (LOC,NR,NC)
      DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),A(1),KA(1)
      COMMON FMT1,FMT2,A
      EQUIVALENCE (FMT1,KFMT1),(FMT2,KFMT2),(A,KA)
   11 KEND=(LOC-1)+(NR*NC)
   12 DO  13   I=LOC,KEND
   13 A(I)=0.0
   14 RETURN
   15 END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      COMMON/INP/X,Y,W,NOV,NSTAGE
      COMMON/BUF/IN(8,2000),KN
      EQUIVALENCE (KAVE1,AVE1),(KAVE2,AVE2)
      DIMENSION NSTG(200),KLO(200),KHI(200),NMERG(200),KASTRK(11,11),
     1          LINE11(11,11),LINE12(11,11),LINE13(11,11),
     2          LINE21(11,11),LINE22(11,11),LINE23(11,11),
     3          LINE31(11,11),LINE32(11,11),LINE33(11,11),
     4          KSPACE(11,11),KSTDY(2),KSTUDY(2),HEADNG(14,3)
      DIMENSION SLIN11(11),SLIN13(11),SLIN21(11),SLIN22(11),SLIN23(11),
     1          SLIN31(11),SLIN33(11)
      DIMENSION IASTRK(11)
      DATA IASTRK/11*1H*/
      DATA IBLANK/6H      /,JASTRK/6H******/,MINUS/1H-/,JPROG/6HDIAGRM/
      INTEGER W
      INTEGER SLIN11,SLIN12,SLIN13,SLIN21,SLIN22,SLIN23,
     1        SLIN31,SLIN32,SLIN33
C
C
      CALL SETUP(10,0,0,0,0,0,20,21,0,0,0,JSEQ)
      CALL NOPAGE
      INP = 10
      IT = 21
      REWIND INP
       IPAGES=0
      JPAGE=1
      IPAGE=1
      READ(5,500) KPRG,KSTDY(1),KSTDY(2),NT
  500 FORMAT(A6,1X,A6,A2,I2)
   10 CONTINUE
      READ(INP) KSTUDY(1), KSTUDY(2), NCASE
      NSTAGE = NCASE-1
      IF(KSTDY(1).EQ.KSTUDY(1).AND.KSTDY(2).EQ.KSTUDY(2)) GO TO 20
      WRITE(6,502) KSTDY(1),KSTDY(2),KSTUDY(1),KSTUDY(2)
  502 FORMAT(21H0INCORRECT STUDY ID ',A6,A2,34H' DOES NOT MATCH GRMBRS S
     1TUDY ID ',A6,A2,17H'. PROGRAM ABORT. )
      STOP
   20 CONTINUE
      IF(NT.LE.3) GO TO 30
      WRITE(6,503)
  503 FORMAT(54H0NUMBER OF TITLE CARDS CANNOT EXCEED 3. PROGRAM ABORT. )
      STOP
   30 CONTINUE
      IF(NT.EQ.0) GO TO 50
      DO 40 JJ=1,NT
   40 READ(5,504) (HEADNG(II,JJ),II=1,14)
  504 FORMAT(13A6,A2)
   50 CONTINUE
      READ(5,505) X,Y,W,NOV,IMIN
  505 FORMAT(2F5.1,2I5,I2)
C
```

LINE # TEXT...

```
C     SORT INPUT -GRMBRS- TAPE INTO ASCENDING KPATH ORDER.
      REWIND IT
      DO 101 I = 1, NSTAGE
      READ(INP) (IN(J,I),J=1,5),D,D,IN(6,I),D,IN(7,I),D,D,D,IN(8,I)
  101 CONTINUE
      KN = 0
      DO 104 I = 1, NSTAGE
      JMIN = 1
      MIN = 2000
      DO 102 J = 1, NSTAGE
      IF(IN(8,J).LT.0) GO TO 102
      IF(IN(8,J).GT.MIN) GO TO 102
      MIN = IN(8,J)
      JMIN = J
  102 CONTINUE
      IF(IN(2,JMIN).LT.W) GO TO 103
      WRITE(IT) (IN(L,JMIN),L=1,8)
      KN = KN + 1
  103 IN(8,JMIN) = -IN(8,JMIN)
  104 CONTINUE
      DO 105 I = 1, NSTAGE
  105 IN(8,I) = IABS(IN(8,I))
      REWIND INP
      REWIND IT
C
C
C     COMPUTE STARTER GROUPS. OUTPUT IS...NSTG - STAGE NUMBER
C                                         KLO  - LOW KPATH SEQUENCE
C                                         KHI  - HIGH KPATH SEQUENCE
C                                         NGRPS- NUMBER OF GROUPS FOUND
C
      CALL START(NSTG,KLO,KHI,NGRPS,IMIN)
      NPAGEJ=NGRPS/11
      IF(MOD(NGRPS,11).NE.0) NPAGEJ=NPAGEJ+1
C
C     INITIALIZATION BEFORE EACH PAGE
C
   60 CONTINUE
      DO 70 II=1,11
      DO 70 JJ=1,11
      KSPACE(II,JJ)=IBLANK
      LINE11(II,JJ)=IBLANK
      LINE12(II,JJ)=IBLANK
      LINE13(II,JJ)=IBLANK
      LINE21(II,JJ)=IBLANK
      LINE22(II,JJ)=IBLANK
      LINE23(II,JJ)=IBLANK
      LINE31(II,JJ)=IBLANK
      LINE32(II,JJ)=IBLANK
      LINE33(II,JJ)=IBLANK
```

LINE #   TEXT...

```
   70 KASTRK(II,JJ)=IBLANK
      KLFT=II*(JPAGE-1)
      KRGT=II*JPAGE
      KTOP=II*(IPAGE-1)
      KBOT=II*IPAGE
      KSTOP=MINO(KRGT,NGRPS)
      KK=0
      DO 80 JJ=1,NGRPS
   80 NMERG(JJ)=0
C
C     PICK UP NEXT GROUP FROM BUFFER SORTED BY -SORT-
C
      KNT=0
  100 CONTINUE
      KNT=KNT+1
      NST=IN(1,KNT)
      NMBRS=IN(2,KNT)
      KAVE1=IN(3,KNT)
      KAVE2=IN(4,KNT)
      NMBR1=IN(5,KNT)
      NMBR2=IN(7,KNT)
      KPATH1=IN(6,KNT)
      KPATH2=IN(8,KNT)
      IF(NST.NE.NSTG(KK+1)) GO TO 200
C
C     STARTER GROUP WAS READ
C
      KK=KK+1
      KLO(KK)=KPATH1
      KHI(KK)=KPATH2
      IF(KK.LE.KLFT.OR.KK.GT.KRGT) GO TO 100
C
C     STORE 3 LINES OF INFORMATION FOR STARTER GROUP
C
      MC=KK-KLFT
      CALL BINBCD(   NST,SLIN11(MC),4,$199,0)
      SLIN12=IBLANK
      CALL BINBCD( NMBRS,SLIN13(MC),4,$199,0)
      CALL BINBCD(KPATH1,SLIN21(MC),4,$199,0)
      SLIN22(MC)=MINUS
      CALL BINBCD(KPATH2,SLIN23(MC),4,$199,0)
      SLIN31(MC)=I2FA(AVE1)
      SLIN32=IBLANK
      SLIN33(MC)=I2FA(AVE2)
      GO TO 100
  199 WRITE(6,198)
  198 FORMAT('1UNABLE TO FIT BINARY VALUE IN SPECIFIED NO. CHARACTERS')
      STOP
C
C     FIND COLUMN(S) INVOLVED IN MERGER ACTION
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
  200 CONTINUE
      IF(KPATH1.GT.KLO(KK)) GO TO 100
      DO 210 JJ=1,KK
      MCL=JJ
  210 IF(KPATH1.LE.KLO(JJ)) GO TO 211
  211 CONTINUE
      DO 220 JJ=1,NGRPS
      MCR=NGRPS-JJ+1
      IF(NMERG(MCR).LT.0) GO TO 220
      IF(KPATH2.GE.KHI(MCR)) GO TO 221
  220 CONTINUE
  221 CONTINUE
      NMERG(MCL)=NMERG(MCL)+1
      KLO(MCL)=KPATH1
      KHI(MCL)=KPATH2
      IF(MCL.LE.KLFT.OR.MCL.GT.KRGT) GO TO 300
      IF(NMERG(MCL).LE.KTOP.OR.NMERG(MCL).GT.KBOT) GO TO 300
      MCOL=MCL-KLFT
      MROW=NMERG(MCL)-KTOP
C
C     STORE 3 LINES OF INFORMATION FOR GROUP IN 'MCL' COLUMN.
C
      CALL BINBCD(   NST,LINE11(MROW,MCOL),4,$199,0)
      CALL BINBCD( NMBRS,LINE13(MROW,MCOL),4,$199,0)
      CALL BINBCD(KPATH1,LINE21(MROW,MCOL),4,$199,0)
      LINE22(MROW,MCOL)=MINUS
      CALL BINBCD(KPATH2,LINE23(MROW,MCOL),4,$199,0)
      LINE31(MROW,MCOL)=I2FA(AVE1)
      LINE33(MROW,MCOL)=I2FA(AVE2)
      KASTRK(MROW,MCOL)=JASTRK
       IF(NMBR1.LT.W.OR.NMBR2.LT.W) GO TO 350
C
C     TWO COLUMNS MERGING TO LEFT. STORE ASTERISKS IN 'MCR' COLUMN.
C
  300 CONTINUE
      IF(MCL.EQ.MCR) GO TO 350
      IF(NMERG(MCR).GT.KBOT) GO TO 320
      IF(NMERG(MCL).LE.KTOP) GO TO 320
      IF(MCR.GT.KSTOP.OR.MCR.LE.KLFT) GO TO 320
      MCOL=MCR-KLFT
      IB=MAX0(NMERG(MCR)-KTOP+1,1)
      IE=MIN0(NMERG(MCL)-KTOP,11)
      DO 310 JJ=IB,IE
      LINE12(JJ,MCOL)=JASTRK
      LINE22(JJ,MCOL)=JASTRK
      LINE32(JJ,MCOL)=JASTRK
  310 KASTRK(JJ,MCOL)=JASTRK
      IF(NMERG(MCL).GT.KBOT.AND.IE.EQ.11) GO TO 320
      LINE32(IE,MCOL)=IBLANK
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      KASTRK(IE,MCOL)=IBLANK
C
C     STORE ASTERISKS IN 'MROW' ROW
C
  320 CONTINUE
      IF(NMERG(MCL).GT.KBOT.OR.NMERG(MCL).LE.KTOP) GO TO 340
      MROW=NMERG(MCL)-KTOP
      IF(MCL.EQ.KSTOP) KSPACE(MROW,11)=JASTRK
      IF(MCL.GE.KSTOP.OR.MCR.LE.KLFT) GO TO 340
      IB=MAXO(MCL-KLFT+1,1)
      IE=MINO(MCR-KLFT,11)
      IF(IB.GT.1) KSPACE(MROW,IB-1)=JASTRK
      DO 330 JJ=IB,IE
      KSPACE(MROW,JJ)=JASTRK
      LINE21(MROW,JJ)=JASTRK
      LINE22(MROW,JJ)=JASTRK
  330 LINE23(MROW,JJ)=JASTRK
      IF(MCR.GT.KSTOP.AND.IE.EQ.11) GO TO 340
      LINE23(MROW,IE)=IBLANK
      KSPACE(MROW,IE)=IBLANK
C
C     FLAG 'MCR' COLUMN INACTIVE
C
  340 CONTINUE
      NMERG(MCR)=-NMERG(MCL)
C
C     CHECK FOR ACTIVE COLUMNS TO LEFT
C
  350 CONTINUE
      DO 355 JJ=1,MCL
      IF(NMERG(JJ).LT.0) GO TO 355
      IF(NMERG(JJ).GE.NMERG(MCL)) GO TO 355
      NMERG(JJ)=NMERG(MCL)
      IF(NMERG(JJ).GT.KBOT.OR.NMERG(JJ).LE.KTOP) GO TO 355
      IF(NMERG(MCL).GT.KBOT.OR.NMERG(MCL).LE.KTOP) GO TO 355
      IF(JJ.LE.KLFT.OR.JJ.GT.KRGT) GO TO 355
      MCOL=JJ-KLFT
      MROW=NMERG(MCL)-KTOP
      LINE12(MROW,MCOL)=JASTRK
      LINE22(MROW,MCOL)=JASTRK
      LINE32(MROW,MCOL)=JASTRK
      KASTRK(MROW,MCOL)=JASTRK
  355 CONTINUE
C
C     CHECK FOR PRINT ON FULL PAGE
C
      IF(NST.EQ.1) GO TO 400
      IF(NMERG(MCL).LT.KBOT) GO TO 100
      IF(KPATH2.LT.KHI(KSTOP)) GO TO 100
C
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C      PRINT 'IPAGE,JPAGE' PAGE
C
  400 CONTINUE
      IF(IPAGE.GT.1) GO TO 435
      WRITE(6,506) JPAGE
  506 FORMAT(1H1,/////,8H COLUMN ,I2)
      IF(NT.EQ.0) GO TO 420
      DO 410 JJ=1,NT
  410 WRITE(6,507) (HEADNG(II,JJ),II=1,14)
  507 FORMAT(36X,13A6,A2)
  420 CONTINUE
      WRITE(6,600) X,Y,W,NOV,NSTAGE
  600 FORMAT(36X,4HX = ,F6.2,5H Y = ,F6.2,4H W =,I2,6H N/V =,I5,
     19H NSTAGE =,I5)
      WRITE(6,508)
  508 FORMAT(///)
      IE=KSTOP-KLFT
      WRITE(6,509) (SLIN11(II),SLIN12     ,SLIN13(II),II=1,IE)
      WRITE(6,509) (SLIN21(II),SLIN22(II),SLIN23(II),II=1,IE)
      WRITE(6,509) (SLIN31(II),SLIN32     ,SLIN33(II),II=1,IE)
      WRITE(6,511) (IASTRK(II),II=1,IE)
      WRITE(6,511) (IASTRK(II),II=1,IE)
  435 CONTINUE
      DO 440 II=1,11
      WRITE(6,509) (LINE11(II,JJ),LINE12(II,JJ),LINE13(II,JJ),JJ=1,11)
      WRITE(6,510) (LINE21(II,JJ),LINE22(II,JJ),LINE23(II,JJ),
     1 KSPACE(II,JJ),JJ=1,11)
      WRITE(6,509) (LINE31(II,JJ),LINE32(II,JJ),LINE33(II,JJ),JJ=1,11)
  509 FORMAT(11(1X,A4,A1,A4,2X))
  510 FORMAT(1X,10(A4,A1,A4,A3),A4,A1,A4,A2)
      WRITE(6,511) (KASTRK(II,JJ),JJ=1,11)
      WRITE(6,511) (KASTRK(II,JJ),JJ=1,11)
  511 FORMAT(11(5X,A1,6X))
  440 CONTINUE
C
C      PAGE CONTROL COMPUTATION
C
      IPAGE=IPAGE+1
      IF(NST.NE.1) GO TO 60
      IF(JPAGE.EQ.NPAGEJ) GO TO 460
       IPAGES=IPAGE
      IPAGE=1
      JPAGE=JPAGE+1
      GO TO 60
  460  IF(IPAGES.EQ.0) GO TO 470
       IF(IPAGES.NE.IPAGE) GO TO 60
  470 CALL MARGIN
      STOP
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE START(NSTG,KLO,KHI,NGRPS,IMIN)
      COMMON/BUF/INN(8,2000),K
      COMMON/INP/X,Y,W,NOV,NSTAG
      INTEGER W
      DIMENSION NSTG(200),KLO(200),KHI(200),IN(8)
      EQUIVALENCE (IN(3),BET)
      NGRPS = 0
      IS = 20
      IT = 21
      DO 10 J=1,8
   10 IN(J)=INN(J,NOV)
   11  IF(IMIN.NE.1) GO TO 12
       IF(BET.LE.X) GO TO 100
       GO TO 13
   12  IF(BET.GE.X) GO TO 100
   13  NOV=NOV+1
      DO 9 J=1,8
    9 IN(J)=INN(J,NOV )
      IF(NOV.EQ.NSTAG) GO TO 1000
      GO TO 11
  100 X=BET
      KREAD = NOV
  111 IF(IN(2).GE.W) GO TO 110
  125 KREAD = KREAD + 1
      IF(KREAD.GT.NSTAG) GO TO 200
      DO 8 J=1,8
    8 IN(J)=INN(J,KREAD)
      GO TO 111
  110 CONTINUE
      IF(NGRPS.EQ.0) GO TO 120
      DO 115 I=1,NGRPS
      IF(IN(6).GE.KLO(I).AND.IN(6).LE.KHI(I)) GO TO 125
      IF(IN(8).GE.KLO(I).AND.IN(8).LE.KHI(I)) GO TO 125
      IF(IN(6).LE.KLO(I).AND.IN(8).GE.KHI(I)) GO TO 125
  115 CONTINUE
  120   CONTINUE
      NGRPS=NGRPS + 1
      IF(NGRPS.GT.200)GO TO 3000
      NSTG(NGRPS) = IN(1)
      KLO(NGRPS)= IN(6)
      KHI(NGRPS) = IN(8)
      GO TO 125
  200 CONTINUE
      NN = NOV - 1
       DO 7 J=1,8
    7 IN(J)=INN(J,NN)
  211  IF(IMIN.NE.1) GO TO 214
       IF(IN(2).GE.W.AND.BET.LE.Y) GO TO 220
       GO TO 215
  214  IF(IN(2).GE.W.AND.BET.GE.Y) GO TO 220
```

PROGRAM TYPE: SUBPROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
  215  NN = NN - 1
       IF(NN.LE.0) GO TO 301
       DO 6 J=1,8
     6 IN(J)=INN(J,NN)
       GO TO 211
  220  CONTINUE
       IF(NGRPS.EQ.0) GO TO 230
       DO 225 I=1,NGRPS
       IF(IN(6).GE.KLO(I).AND.IN(6).LE.KHI(I)) GO TO 215
       IF(IN(8).GE.KLO(I).AND.IN(8).LE.KHI(I)) GO TO 215
       IF(IN(6).LE.KLO(I).AND.IN(8).GE.KHI(I)) GO TO 215
  225  CONTINUE
  230  CONTINUE
       NGRPS = NGRPS + 1
       IF(NGRPS.GT.200) GO TO 3000
        NSTG(NGRPS) = IN(1)
       KLO(NGRPS)=IN(6)
       KHI(NGRPS) = IN( 8)
        GO TO 215
  301  IF(NGRPS.EQ.0) GO TO 2000
        REWIND IS
       DO 300 I=1,NGRPS
       JMIN = 1
       MIN = 2000
       DO 320 J=1,NGRPS
       IF(KHI(J).LT.0) GO TO 320
       IF(KHI(J).GT.MIN) GO TO 320
       MIN = KHI(J)
       JMIN = J
  320   CONTINUE
       WRITE(IS) NSTG(JMIN),KLO(JMIN),KHI(JMIN)
       KHI(JMIN)=-KHI(JMIN)
  300  CONTINUE
       DO 600 I=1,NGRPS
  600  KHI(I)=IABS(KHI(I))
       ENDFILE IS
       REWIND IS
       DO 400 I=1,NGRPS
  400  READ(IS) NSTG(I),KLO(I),KHI(I)
       REWIND IS
       DO 500 J=1,K
  500  READ(IT) (INN(I,J),I=1,8)
       REWIND IT
       RETURN
  1000 WRITE(6,1010)
  1010 FORMAT(38H NO AVE BETWEEN  GE  X  ON INPUT TAPE    )
       STOP
  2000 WRITE(6,2010)
  2010 FORMAT(25H NO STARTER GROUPS FOUND     )
       STOP
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
3000 WRITE(6,3010)
3010 FORMAT(53H TO MANY STARTER GROUPS FOUND, CHANGE INPUT, RESUBMIT  )
     STOP
     END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      FUNCTION I2FA(X)
      IF(X.GE.99.9) X = 99.9
      ENCODE(100,J) X
100   FORMAT(F4.1)
      I2FA = J
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C
C
C
C
C   THIS PROGRAM WILL PRINT A TWO WAY DISTRIBUTION ON A SPECIFIED SAMPLE.
C
C   THE PROGRAM FIRST READS IN CONTROL CARDS AND CHECKS FOR ERRORS,
C
C   THEN IT READS IN THE CODAP TAPE, CHECKS ID AND EXTRACTS THE
C
C   THE REQUIRED INFORMATION PRECEDING THE CASE DATA.  ALL THE CASES
C
C   ARE THEN PASSED THROUGH ONCE FOR EACH DISTRIBUTION WHICH IS DELIMITED
C
C   BY A 'DIST2X' THRU 'END' CARD.
C
C
C   AFTER A CASE IS READ IN, ITS ROW AND COLUMN VALUES ARE DECODED AND
C
C   DETERMINED TO BE VALID OR INVALID NUMERICS.  IF THE DISTRIBUTION
C
C   USES INTERVALS, THE ROW AND COLUMN VALUES ARE DISTRIBUTED INTO THE
C
C   THE PROPER INTERVALS.  IF THE 'OTHER' OPTION WAS NOT SPECIFIED, ALL
C
C   VALUES NOT WITHIN THE SPECIFIED LOW-HIGH RANGE ARE ASSUMED TO BE
C
C   INVALID WHETHER NUMERIC OR NOT.
C
C
C   EACH UNIQUE ROW-COLUMN PAIR IS WRITTEN IN A TABLE IN CORE.  IF THIS
C
C   TABLE OVERFLOWS, IT IS MERGED WITH A DISK FILE.  AFTER ALL CASES ARE
C
C   PROCESSED, THE DISK FILE IS REWOUND AND THE PAIRS ARE READ BACK AND
C
C   ACCUMULATED INTO COMPLETE ROWS.  THESE COMPLETE ROWS ARE THEN
C
C   WRITTEN ONTO ANOTHER DISK FILE.  WHEN THAT IS COMPLETE, THE OTHER
C
C   DISK FILE IS ALSO REWOUND FOR PRINTING.
C
C
C
      INTEGER BLK, FIRST, SAME, HIGH, RTABLE, RUN,DOT
C
      DIMENSION MATA(4),MATB(4),MATC(5),MATD(4),MATE(4),MATF(5)
      DIMENSION MAT(3),IBR(225),BR(1),ITR(225),TR(1),IBC(225),BC(1),
     1ITC(225),TC(1),IREL(1000),IDICT(15,16),LABEL(15,3),IHEAD(14,3),
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
     2C(500),KV(1000),FTABLE(5,1000),ITABLE(5,1),CTABLE(4,450),
     3PAIR(2),RTABLE(450),ISCALE(9300),JTABLE(4,1)
      DIMENSION ITITLE(5),PCT(10),ISTAT(4),CT(5,1)
      DIMENSION IBH(450),ITH(450),BH(1),TH(1)
      COMMON /COMODE/IFLR,IFLC
      EQUIVALENCE (IBH(1),BH(1)) , (ITH(1),TH(1))
C DATA STATEMENTS
      DATA ISAMP,IROW,ICOL,IEND,MAT/6HSAMPLE,3HROW,6HCOLUMN,3HEND,3*1H /
      DATA ITOTAL,IVALD,MEAN,ISD/6H TOTAL,6H VALID,6H  MEAN,6H  S.D./
      DATA IDPRGM,IPLUS,BLK,MEMBER,NOGO/6HDIST2X,1H+,1H ,1H1,4HNOGO/
      DATA IOTHER/6H OTHER/,JA,IA/2H**,1H*/,JTOTAL/3HTOT/
      DATA ISVAL,NOTVAL/6H00C00C,6H0000C/,LIMIT/037777777777777/
      DATA JCOMP, IHIST /1HC,1HV/ , INGR /1HO/,DOT/6H....../
      DATA IALL, ICASES /6H ALL C, 4HASES/
C EQUIVALENCES TO CONSERVE CORE (ISCALE IS FOR THIS PURPOSE ONLY)
      EQUIVALENCE (ISCALE(  46),MAT(1)),(ISCALE(  97),PAIR(1)),
     1(ISCALE(   1),IHEAD(1,1) ),(ISCALE(  51),LABEL(1,1) ),
     2(ISCALE( 101),BR(1)      ),(ISCALE( 101),IBR(1)     ),
     3(ISCALE( 326),TR(1)      ),(ISCALE( 326),ITR(1)     ),
     4(ISCALE( 551),BC(1)      ),(ISCALE( 551),IBC(1)     ),
     5(ISCALE( 776),TC(1)      ),(ISCALE( 776),ITC(1)     ),
     6(ISCALE(1001),IREL(1)    ),(ISCALE(1001),IDICT(1,1) ),
     7(ISCALE(1001),C(1)       ),(ISCALE(1001),RTABLE(1)  ),
     8(ISCALE(1501),KV(1)      ),(ISCALE(2501),FTABLE(1,1)),
     9(ISCALE(2501),ITABLE(1,1)),(ISCALE(7501),CTABLE(1,1))
      EQUIVALENCE (IVALR,PAIR(1)),(VALR,IVALR),(IVALC,PAIR(2)),
     1(IVALC,VALC),(JVAL,ISCALE(47)),(KVAL,ISCALE(48))
      EQUIVALENCE (ISCALE(5001),ITITLE(1)),(ISCALE(5011),PCT(1)),
     1(ISCALE(5025),ISTAT(1)),(ISCALE(2501),CT(1,1))
      EQUIVALENCE (ISCALE(7501),JTABLE(1,1))
      INTEGER COMPA,COMPN,COMPRC,COMPC
      EXTERNAL COMPA, COMPN
      CALL SETUP(10,0,0,0,0,0,20,21,0,0,0,JSEQ)
      INP = 10
    3 INVALR = IPLUS
      INVALC = IPLUS
      KKR = JA
      KKC = JA
      INGRP = INGR
      ISS = 0
      IRS = 0
      ICS = 0
      ISR = 0
      ISC = 0
      INR = 0
      INC = 0
      ISW = 0
      JSW = 0
      KSW = 0
      LSW = 0
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      LIN = 20
      LOU = 21
      NROW = 0
      NCOL = 0
      IPAGE = 0
      LPAGE = 0
      NXROW = 0
      KODER = 0
      KODEC = 0
      IGT = 0
      IVRT = 0
      IVCT = 0
      ROWTOT = 0.0
      COLTOT = 0.0
      RSQTOT = 0.0
      CSQTOT = 0.0
      IXX = 0
      ISHF18 = 2**18
      MAXC = 450
      MAXF = 1000
      MINF = MAXF
      MINC = MAXC
      JDS = 0
      JDR = -1
      JDC = -1
      ISCALE(1) = BLK
      CALL TMTCSC(99,ISCALE(2),ISCALE(1))
      LABEL(2,3) = IALL
      LABEL(3,3) = ICASES
      ASSIGN 103 TO JSAMP
      ASSIGN 104 TO JROW
      ASSIGN 109 TO JCOL
      ASSIGN 114 TO MROW
      ASSIGN 123 TO MCOL
      ASSIGN 189 TO NORMAL
      ASSIGN 300 TO NORM2
      ASSIGN 501 TO IPASS
      ASSIGN 518 TO LPASS
      ASSIGN 723 TO IXROW
      ASSIGN 740 TO JUMRP
      ASSIGN 748 TO JUMCP
      ASSIGN 775 TO JUMTP
      CALL RESET
C
C  READ CONTROL CARD
C
      READ(5,801,END=999) IDPGM, ID1,ID2, NHEAD, RUN, JUMP
      CALL PRINTC
      IF(JUMP.NE.0) ASSIGN 760 TO JUMTP
      IF(RUN.NE.BLK) RUN = NOGO
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

DIST2X TWO WAY DISTRIBUTION PROGRAM

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          IF(IDPGM.NE.IDPRGM) GO TO 901
    700   IF(NHEAD.EQ.0) GO TO 2
          IF(NHEAD.GT.3) GO TO 902
    701   DO 1 J = 1, NHEAD
          READ(5,802)(IHEAD(I,J),I=1,14)
          CALL PRINTC
    1     CONTINUE
    2     CONTINUE
    C
    C  READ   ROW-COLUMN-SAMPLE DEFINITION CARDS
    C
    10    READ(5,803,END=908)KODE,KIND,ISUB,I1,I2,I3,IOP1,IOP2,IOP3,C1,C2,C3
    C
          IF(KODE.EQ.BLK) GO TO 36
          IF(KODE.EQ.IEND) GO TO 42
    C
          IF(KODE.NE.ISAMP) GO TO 11
    C
    C  PROCESS SAMPLE CARD
          CALL PRINTC
          ENCODE(807,IDS) KIND, ISUB
    807   FORMAT(A1,A3,' =')
          JDS = -1
          IF(KIND.EQ.JCOMP) JDS = 1
          IF(KIND.NE.JCOMP.AND.KIND.NE.IHIST) GO TO 924
          CALL BCDBIN( ISUB, ISS,      3,$925)
          ASSIGN 102 TO JSAMP
          INGRP = I1
          IF(I1.EQ.BLK) INGRP = MEMBER
          IF(KIND.EQ.IHIST) GO TO 10
          GROUP = 0.0
          IF(I1.EQ.'1') GROUP = 1.0
          ASSIGN 100 TO JSAMP
          GO TO 10
    C
    11    IF(KODE.NE.IROW) GO TO 28
    C
    C  PROCESS ROW CARD
          CALL PRINTC
          ENCODE(807,IDR) KIND, ISUB
          IF(KIND.EQ.JCOMP) JDR = 1
          IF(IRS.EQ.1) GO TO 915
    714   IRS = 1
          IF(KIND.NE.JCOMP.AND.KIND.NE.IHIST) GO TO 924
          CALL BCDBIN( ISUB, ISR,      3,$925)
          IR1 = IOP1 + 1
          IR2 = IOP2 + 1
          IR3 = IOP3 + 1
          IRET = 1
    C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C  BEGIN TO DETERMINE THE DISTRIBUTION METHOD
C
12     IF(KIND.EQ.JCOMP) GO TO 20
       ITYPE = -1
       IF(I1.NE.BLK.AND.I2.NE.BLK) GO TO 13
       METHOD = 0
       NCHAR = -1
       GO TO (21,29),IRET
C
13     IF(I3.EQ.BLK) GO TO 19
       DO 14 I = 1, 6
       J3 = I3
       NCHAR = 7 - I
       CALL BCDBIN(   J3,  J3, NCHAR,$14)
       GO TO 15
14     CONTINUE
       GO TO 906
15     IF(J3.EQ.0) GO TO 16
       I3 = J3
       CALL BCDBIN(   I1,  I1, NCHAR,$906)
       CALL BCDBIN(   I2,  I2, NCHAR,$906)
       METHOD = 1
       GO TO (21,29),IRET
C
16     METHOD = -1
       DO 17 I = 1, 6
       J2 = I2
       NCHAR = 7 - I
       CALL BCDBIN(   J2,  J2, NCHAR,$17)
       GO TO 18
17     CONTINUE
       GO TO 906
C
18     I2 = J2
       CALL BCDBIN(   I1,  I1, NCHAR,$906)
       GO TO (21,29), IRET
C
19     METHOD = -1
       NCHAR = -1
       GO TO (21,29), IRET
C
20     ITYPE = 1
       METHOD = 1
       IF(C3.EQ.0.0) METHOD = -1
       GO TO (21,29), IRET
C  END METHOD DETERMINATION
C
C
C  ESTABLISH ROW BOUNDARIES IF INTERVALS ARE USED
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
 21    IF(METHOD) 22,10 ,25
 C
 C  IRREGULAR DISTRIBUTION
 22    IF(ITYPE) 23,917,24
 23    IRREG1=2
       KODER = 1
       ASSIGN 724 TO IXROW
       ASSIGN 118 TO MROW
       IF(NCHAR.LE.0) GO TO 38
       KODER = 3
       ASSIGN 724 TO IXROW
       ASSIGN 105 TO JROW
       ASSIGN 119 TO MROW
       IRREG1 = 1
       IRREG2 = 1
       GO TO 38
 24    IRREG1 = 4
       KODER = 5
       ASSIGN 724 TO IXROW
       ASSIGN 107 TO JROW
       ASSIGN 120 TO MROW
       GO TO 40
 C  REGULAR DISTRIBUTION
 25    IF(ITYPE) 26,917,27
 26    MINROW = I1
       MAXROW = I2
       INTROW = I3
       KODER = 2
       ASSIGN 728 TO IXROW
       ASSIGN 105 TO JROW
       ASSIGN 115 TO MROW
       INR = (MAXROW - MINROW)/INTROW  + 1
       GO TO 10
 27    BOTROW = C1
       TOPROW = C2
       ROWINT = C3
       KODER = 4
       ASSIGN 725 TO IXROW
       ASSIGN 107 TO JROW
       ASSIGN 117 TO MROW
       INR = (TOPROW - BOTROW)/ROWINT + 1.0
       GO TO 10
 C
 C PROCESS COLUMN CARDS
 C
 28    IF(KODE.NE.ICOL) GO TO 42
       CALL PRINTC
       ENCODE(807,IDC) KIND, ISUB
       IF(KIND.EQ.JCOMP) JDC = 1
       IF(ICS.EQ.1) GO TO 916
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
 715    ICS = 1
        IF(KIND.NE.JCOMP.AND.KIND.NE.IHIST) GO TO 924
        CALL BCDBIN( ISUB, ISC,      3,$925)
        IC1 = IOP1 + 1
        IC2 = IOP2 + 1
        IC3 = IOP3 + 1
        IRET = 2
        GO TO 12
C
C  ESTABLISH COLUMN BOUNDARIES IF INTERVALS ARE USED
 29     IF(METHOD) 30,10 ,33
C
C  IRREGULAR DISTRIBUTION
 30     IF(ITYPE) 31,917,32
 31     IRREG1 = 3
        KODEC = 1
        ASSIGN 504 TO JPASS
        ASSIGN 127 TO MCOL
        IF(NCHAR.LE.0) GO TO 39
        IRREG1 = 1
        IRREG2 = 2
        KODEC = 3
        ASSIGN 504 TO JPASS
        ASSIGN 110 TO JCOL
        ASSIGN 128 TO MCOL
        GO TO 39
C
 32     IRREG1 = 5
        KODEC = 5
        ASSIGN 504 TO JPASS
        NUMC = ISVAL
        ASSIGN 112 TO JCOL
        ASSIGN 129 TO MCOL
        GO TO 41
C
C  REGULAR DISTRIBUTION
 33     IF(ITYPE) 34,917,35
 34     MINCOL = I1
        MAXCOL = I2
        INTCOL = I3
        KODEC = 2
        ASSIGN 506 TO JPASS
        ASSIGN 110 TO JCOL
        ASSIGN 124 TO MCOL
        INC = (MAXCOL - MINCOL)/INTCOL + 1
        GO TO 10
 35     BOTCOL = C1
        TOPCOL = C2
        COLINT = C3
        KODEC = 4
```

DIST2X TWO WAY DISTRIBUTION PROGRAM

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

```
LINE #   TEXT...
            ASSIGN 505 TO JPASS
            ASSIGN 112 TO JCOL
            ASSIGN 126 TO MCOL
            INC = (TOPCOL - BOTCOL)/COLINT  + 1.0
            GO TO 10
      C
      C
      C  SET UP THE IRREGULAR INTERVALS IF REQUIRED
      C
      36    CALL PRINTC
            IF(METHOD.NE.(-1)) GO TO 918
            GO TO (37,38,39,40,41), IRREG1
      C
      37    CALL BCDBIN(   I1,  I1, NCHAR,$906)
            CALL BCDBIN(   I2,  I2, NCHAR,$906)
      C
            GO TO (38,39), IRREG2
      C
      38    INR = INR + 1
            IBR(INR) = I1
            ITR(INR) = I2
            GO TO 10
      C
      39    INC = INC + 1
            IBC(INC) = I1
            ITC(INC) = I2
            GO TO 10
      C
      40    INR = INR + 1
            BR(INR) = C1
            TR(INR) = C2
            GO TO 10
      41    INC = INC + 1
            BC(INC) = C1
            TC(INC) = C2
            GO TO 10
      C
      C PROCESS END CARD
      42    CALL PRINTC
      C
      C  PERFORM ERROR CHECKS
            IF(KODE.NE.IEND) GO TO 919
      704   CONTINUE
            JSW = 1
            IF(IMS.NE.1) GO TO 903
      702   CONTINUE
            IF(ICS.NE.1) GO TO 904
      703   CONTINUE
            IF(ISS.EQ.0) IDS = IPLUS
            IF(MOD(KODEC,2).EQ.0) GO TO 711
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      IF(INC.GT.225) GO TO 905
711   IF(MOD(KODER,2).EQ.0) GO TO 712
      IF(INR.GT.225) GO TO 905
709   CONTINUE
712   IF(RUN.NE.NOGO) GO TO 713
      IF(INC.GT.450) CALL ERRMSG(6,'. COLUMNS MAY EXCEED 450 AND ABORT')
713   CONTINUE
C
C
C  STOP IF NOGO WAS SPECIFIED
      IF(RUN.EQ.NOGO) GO TO 912
      REWIND INP
      REWIND LIN
      REWIND LOU
      WRITE(LIN) (LIMIT,I=1,5)
      ENDFILE LIN
      REWIND LIN
C
      IFLR = 0
      IFLC = 0
      IF(KODER.EQ.0) IFLR = 1
      IF(KODEC.EQ.0) IFLC = 1
      IF(KODER.EQ.0) GO TO 716
      INVALR = INR + IR3
716   IF(KODEC.EQ.0) GO TO 717
      INVALC = INC + IC3
717   CONTINUE
      IF(KODEC.NE.0) ASSIGN 502 TO IPASS
      NSTAT = 2
      IF(IR1.EQ.2) NSTAT = 4
      IF(IR2.EQ.2) ASSIGN 733 TO JUMRP
      IF(IC2.EQ.2) ASSIGN 741 TO JUMCP
      NLR = IR2 + IC2 + JUMP
      IF(KODER.GT.3) NLR = NLR + 1
      IF(IC3.EQ.2.OR.KODEC.EQ.0) KKR = IA
      IF(IR3.EQ.2.OR.KODER.EQ.0) KKC = IA
      KY = KODEC/2 + 1
C
C
C
C
C  OBTAIN DATA FROM THE COMMUNICATIONS REGION OF THE INPUT TAPE
      READ(INP) N, IDT1, IDT2, NCASE, NDICT
C
C  CHECK TAPE-ID
      IF(IDT1.NE.ID1.OR.IDT2.NE.ID2) GO TO 921
C
C  READ THE RELATIVE LOCATION TABLE FROM THE INPUT TAPE
      READ(INP) N, (IREL(I),I=1,N)
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C  DETERMINE THE RELATIVE LOCATIONS FOR HISTORY (VXXX) VARIABLES
       IF(JDR.GT.0) GO TO 43
       J = IREL(ISR)/ISHF18
       IF(J.NE.1) GO TO 923
       LR = MOD(IREL(ISR),ISHF18)
C
43     IF(JDC.GT.0) GO TO 44
       J = IREL(ISC)/ISHF18
       IF(J.NE.1) GO TO 923
       LC = MOD(IREL(ISC),ISHF18)
C
44     IF(JDS.GE.0) GO TO 45
       J = IREL(ISS)/ISHF18
       IF(J.NE.1) GO TO 923
       LS = MOD(IREL(ISS),ISHF18)
C
C  FIND THE DICTIONARY ENTRIES FOR ALL VARIABLES USED
C
45     DO 51 K = 1, NDICT
C
C  READ DICTIONARY BLOCK FROM INPUT TAPE
C
       READ(INP) ND, ((IDICT(I,J),I=1,15),J=1,ND)
C
       DO 51 J = 1, ND
C
       IF(IDR.NE.IDICT(1,J))GO TO 47
       DO 46 I = 1, 14
46     LABEL(I,1) = IDICT(I,J)
       CALL FORMAT(IDICT(14,J),MAT,NFCHAR,IFTYPE,IFLD1,IRCHAR,$920)
C
47     IF(IDC.NE.IDICT(1,J)) GO TO 49
       DO 48 I = 1, 14
48     LABEL(I,2) = IDICT(I,J)
       CALL FORMAT(IDICT(14,J),MAT,NFCHAR,IFTYPE,IFLD1,ICCHAR,$920)
C
49     IF(IDS.NE.IDICT(1,J)) GO TO 51
       DO 50 I = 1, 14
50     LABEL(I,3) = IDICT(I,J)
       CALL FORMAT(IDICT(14,J),MAT,NFCHAR,IFTYPE,IFLD1,ISCHAR,$920)
51     CONTINUE
C
C  PERFORM ERROR CHECKS
C
       IF(LABEL(1,1).NE.IDR) GO TO 911
       IF(LABEL(1,2).NE.IDC) GO TO 910
       IF(ISS.EQ.0) GO TO 705
       IF(LABEL(1,3).NE.IDS) GO TO 909
705    CONTINUE
C
```

LINE #   TEXT...

```
C
C
C
C INITIALIZE COUNTERS
C
      ROW = 0.0
      COL = 0.0
      RSQ = 0.0
      CSQ = 0.0
      RSUM = 0.0
      CSUM = 0.0
      IGNORE = 0
      IUSED = 0
C
C
C
C
C
C
C BEGIN CASE DATA LOOP
C
      DO 349 INDEX = 1, NCASE
C
C  READ IN ANOTHER CASE
      READ(INP) NC, (C(I),I=1,NC), NV,(KV(I),I=1,NV)
C
C
C
      GO TO JSAMP, (100,102,103)
C
C  DETERMINE IF THE CASE IS IN THE SAMPLE
100   IF(C(ISS).EQ.GROUP) GO TO 103
C
C  IF NOT, SKIP IT BY READING IN ANOTHER CASE
101   IGNORE = IGNORE + 1
      GO TO 349
C
102   IF(KV(LS).NE.INGRP) GO TO 101
C
C
C
C  DECODE THE ROW VARIABLE, CHECK VALIDITY, AND RECORD THE SUMS
C
103   GO TO JROW, (104,105,107)
C
104   KROW = KV(LR)
      ROW = 0.0
      NUMR = NOTVAL
      CALL BCDBIN( KROW,   J,IRCHAR,$108)
      ROW = J
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      NUMR = ISVAL
      GO TO 108
C
105   KROW = KV(LR)
      CALL BCDBIN( KROW,KROW,IRCHAR,$106)
      NUMR = ISVAL
      ROW = KROW
      GO TO 108
C
106   KROW = IPLUS
      ROW = 0.0
      NUMR = NOTVAL
      GO TO 108
C
107   ROW = C(ISR)
      NUMR = ISVAL
      IF(ROW.NE.(-99999999.0)) GO TO 108
      NUMR = NOTVAL
C
C
C
C
C  DECODE THE COLUMN VARIABLE, CHECK VALIDITY, AND RECORD SUMS
108   GO TO JCOL,(109,110,112)
109   KCOL = KV(LC)
      COL = 0.0
      NUMC = NOTVAL
      CALL BCDBIN( KCOL,   J,ICCHAR,$113)
      NUMC = ISVAL
      COL = J
      GO TO 113
110   KCOL = KV(LC)
      CALL BCDBIN( KCOL,KCOL,ICCHAR,$111)
      NUMC = ISVAL
      COL = KCOL
      GO TO 113
111   KCOL = IPLUS
      COL = 0.0
      NUMC = NOTVAL
      GO TO 113
112   COL = C(ISC)
      NUMC = ISVAL
      IF(COL.NE.(-99999999.0)) GO TO 113
      NUMC = NOTVAL
C
C
C  DETERMINE THE INTERVAL IF REQUIRED FOR ROW
113   GO TO MROW, (114,115,117,118,119,120)
C
C  RAW DISTRIBUTION
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C
114    IVALR = KROW
       GO TO 122
C
C  REGULAR DISTRIBUTION
C
115    IF(KROW.LT.MINROW.OR.KROW.GT.MAXROW) GO TO 116
       IVALR = (KROW - MINROW) / INTROW  + 1
       GO TO 122
116    IVALR = INR + 1
       IF(ROW.EQ.(-99999999.0)) ROW = 0.0
       GO TO 122
117    IF(ROW.LT.BOTROW.OR.ROW.GT.TOPROW) GO TO 116
       IVALR = (ROW - BOTROW) / ROWINT  +1.0
       GO TO 122
C
C  IRREGULAR DISTRIBUTION
C
118    CALL DISTX1(INR,IBR,ITR,KROW,COMPA,ISUB)
       GO TO 121
119    CALL DISTX1(INR,IBR,ITR,KROW,COMPN,ISUB)
       GO TO 121
120    CALL DISTX1(INR,BR,TR,ROW,COMPN,ISUB)
       IF(ROW.EQ.(-99999999.0)) ROW = 0.0
121    IF(ISUB.EQ.0) ISUB = INR + 1
       IVALR = ISUB
C
C  DETERMINE THE INTERVAL IF REQUIRED FOR COLUMN
C
122    GO TO MCOL, (123,124,126,127,128,129)
C
C  RAW DISTRIBUTION
123    IVALC = KCOL
       GO TO 131
C
124    IF(KCOL.LT.MINCOL.OR.KCOL.GT.MAXCOL) GO TO 125
       IVALC = (KCOL - MINCOL) / INTCOL + 1
       GO TO 131
125    IVALC = INC + 1
       IF(COL.EQ.(-99999999.0)) COL = 0.0
       GO TO 131
126    IF(COL.LT.BOTCOL.OR.COL.GT.TOPCOL) GO TO 125
       IVALC = (COL - BOTCOL) / COLINT + 1.0
       GO TO 131
C
127    CALL DISTX1(INC,IBC,ITC,KCOL,COMPA,ISUB)
       GO TO 130
128    CALL DISTX1(INC,IBC,ITC,KCOL,COMPN,ISUB)
       GO TO 130
129    CALL DISTX1(INC,BC,TC,COL,COMPN,ISUB)
```

LINE # TEXT...

```
      IF(COL.EQ.(-99999999.0)) COL = 0.0
130   IF(ISUB.EQ.0) ISUB = INC+1
      IVALC = ISUB
C
C
C
C
131   IUSED = IUSED + 1
C
C
C  THE FOLLOWING TRANSFER NORMALLY GOES TO STATEMENT 200
C       IF FTABLE IS EMPTY, IT TRANSFERS TO STATEMENT 189
C       IF FTABLE HAS ONE UNIQUE ENTRY GO TO STATEMENT 191
C       (NOTE THAT SOME SECTIONS BEFORE 200 ARE GENERAL OPERATIONS)
C
      GO TO NORMAL, (200,189,191)
C
C     RECORD FIRST ENTRY IN FTABLE
C
189   ASSIGN 191 TO NORMAL
      MINF = MAXF
      MID = MAXF
C
C  UNIVERSAL 'INSERT ENTRY' SECTION
C
190   ITABLE(1,MID) = IVALR
      ITABLE(2,MID) = IVALC
      ITABLE(3,MID) = NUMC
      FTABLE(4,MID) = COL
      FTABLE(5,MID) = COL * COL
      GO TO NORM2, (310,300,302)
C
C     RECORD SECOND UNIQUE ENTRY IN FTABLE
C
191   MID = MAXF
      ASSIGN 200 TO NORMAL
      IF(COMPRC(IVALR,ITABLE(1,MID))) 192,194,193
C
C  UNIVERSAL 'GREATER THAN FTABLE(1,MID)' TRANSMIT SECTION
C
192   N = (MID - MINF) * 5
      MINF = MINF - 1
      IF(MINF.GE.1) GO TO 197
      CALL OVRFLO(MINF,MAXF,N,LIMIT,KSW,ITABLE,LIN,LOU)
      GO TO 189
197   CALL TMTCSC(N,FTABLE(1,MINF),FTABLE(1,MINF+1))
      MID = MID - 1
      GO TO 190
C
C  UNIVERSAL 'LESS THAN FTABLE(1,MID)' TRANSMIT SECTION
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND


```
C
193   N = (MID - MINF + 1) * 5
      MINF = MINF - 1
C  IF CORE TABLE OVERFLOWS, MERGE CORE WITH DISK UNIT
      IF(MINF.GE.1) GO TO 199
      CALL OVRFLO(MINF,MAXF,N,LIMIT,KSW,ITABLE,LIN,LOU)
      GO TO 189
199   CALL TMTCSC(N,FTABLE(1,MINF),FTABLE(1,MINF+1))
      GO TO 190
C
C  IF THIS ENTRY IS NOT DIFFERENT, RETURN TO ABOVE SECTION NEXT TIME
C
194   ASSIGN 191 TO NORMAL
C
C  UNIVERSAL 'EQUAL TO FTABLE(1,MID)' ACCUMULATE SECTION
C
195   ITABLE(3,MID) = ITABLE(3,MID) + NUMC
      FTABLE(4,MID) = FTABLE(4,MID) + COL
      FTABLE(5,MID) = FTABLE(5,MID) + COL * COL
      GO TO NORM2, (310,300,302)
C
C
C  SEARCH CORE TABLE (FTABLE = ITABLE) FOR MATCHING ROW-COLUMN ENTRY
C
C  NORMAL BINARY LOOK-UP SECTION
C
200   FIRST = MAXF
      LAST = MINF
C
C     BEGIN LOOP
C
201   MID = (FIRST + LAST + 1)/2
      IF(MID.EQ.FIRST) GO TO 204
      IF(COMPRC(IVALR,ITABLE(1,MID))) 202,195,203
202   FIRST = MID
      GO TO 201
203   LAST = MID
      GO TO 201
C     END SEARCH LOOP
C
C
C  ENTRY IS FIRST,FIRST-1,OR NOT IN TABLE (INSERT IF NOT IN FTABLE)
204   IF(COMPRC(IVALR,ITABLE(1,MID))) 205,195,193
205   MID = LAST
      IF(COMPRC(IVALR,ITABLE(1,MID))) 192,195,193
C  END OF BINARY LOOK-UP/INSERT
C
C
C
C  UPDATE COLUMN TABS
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
300   ASSIGN 302 TO NORM2
      MINC = MAXC
      MIDC = MAXC
301   JTABLE(1,MIDC) = IVALC
      NCOL = NCOL + 1
      IF(IVALR.EQ.INVALR) GO TO 721
      CTABLE(2,MIDC) = ROW
      CTABLE(3,MIDC) = ROW * ROW
      JTABLE(4,MIDC) = NUMR
      GO TO 349
C
721   CTABLE(2,MIDC) = 0.0
      CTABLE(3,MIDC) = 0.0
      JTABLE(4,MIDC) = NOTVAL
      GO TO 349
C
302   MIDC = MAXC
      ASSIGN 310 TO NORM2
      IF(COMPC(IVALC,JTABLE(1,MIDC))) 303,305,304
C
303   N = (MIDC - MINC) * 4
      MINC = MINC - 1
      IF(MINC.LT.1) GO TO 905
      CALL TMTCSC(N,CTABLE(1,MINC),CTABLE(1,MINC+1))
      MIDC = MIDC - 1
      GO TO 301
C
304   N = (MIDC - MINC + 1) * 4
      MINC = MINC - 1
      IF(MINC.LT.1) GO TO 905
      CALL TMTCSC(N,CTABLE(1,MINC),CTABLE(1,MINC+1))
      GO TO 301
C
305   ASSIGN 302 TO NORM2
C
C  RECORD COLUMN ENTRIES AND SUMS
C
306   IF(IVALR.EQ.INVALR) GO TO 720
      CTABLE(2,MIDC) = CTABLE(2,MIDC) + ROW
      CTABLE(3,MIDC) = CTABLE(3,MIDC) + ROW * ROW
      JTABLE(4,MIDC) = JTABLE(4,MIDC) + NUMR
      GO TO 349
C
720   JTABLE(4,MIDC) = JTABLE(4,MIDC) + NOTVAL
      GO TO 349
C
C
310   HIGH = MAXC
      LOW = MINC
```

LINE #   TEXT...

```
C
311   MIDC = (HIGH + LOW + 1) / 2
      IF(MIDC.EQ.HIGH) GO TO 314
      IF(COMPC(IVALC,JTABLE(1,MIDC)))312,306,313
312   HIGH = MIDC
      GO TO 311
C
313   LOW = MIDC
      GO TO 311
C
314   IF(COMPC(IVALC,JTABLE(1,MIDC))) 315,306,304
315   MIDC= LOW
      IF(COMPC(IVALC,JTABLE(1,MIDC))) 303,306,304
C  END OF COLUMN UPDATE
349   CONTINUE
C  END OF CASE DATA LOOP
C  MERGE CORE TABLE WITH DISK TO PICK UP REMAINING CASES
      MINF = MINF - 1
      CALL OVRFLO(MINF,MAXF,N,LIMIT,KSW,ITABLE,LIN,LOU)
C
C  DISK NOW CONTAINS ALL ROW-COLUMN PAIRS
350   CONTINUE
      REWIND INP
      READ(LIN) IVALR,IVALC,ISP,R,RSQ
      GO TO 353
C
C  ACCESS NEXT ROW-COLUMN DATA POINT
351   READ(LIN) IVALR,IVALC,ISP,R,RSQ
      IF(IVALR.EQ.SAME) GO TO 354
      IGT = IGT + ITOT
      ROWTOT = ROWTOT + SUM
      RSQTOT = RSQTOT + SUMSQ
      IVRT = IVRT + IRVAL
      RVAL = IRVAL
      IF(IRVAL.EQ.0) GO TO 352
      RMEAN = SUM/RVAL
      RSD = SQRT(ABS(SUMSQ/RVAL - RMEAN * RMEAN))
352   WRITE(LOU) SAME,IRVAL,ITOT,RMEAN,RSD,(RTABLE(I),I=1,NCOL)
353   IRVAL = 0
      RMEAN = 0.0
      RSD = 0.0
      IF(IVALR.EQ.LIMIT) GO TO 361
      ITOT = 0
      CALL ZEROI(RTABLE,450)
      SUMSQ = 0.0
      SUM = 0.0
      NROW = NROW + 1
      SAME = IVALR
C
C  ACCUMULATE DATA FOR THIS ROW
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
354   IFREQ = FLD(18,18,ISP)
      ITOT = ITOT + IFREQ
      IF(IVALC.EQ.INVALC) GO TO 351
      IVAL = FLD( 0,18,ISP)
      IRVAL = IRVAL + IVAL
      SUM = SUM + R
      SUMSQ = SUMSQ + RSQ
C
C   FIND CORRECT SUBSCRIPT OF CURRENT COLUMN
355   LOW = MINC
      HIGH = MAXC
      ITEST = MAXC + 1
356   MIDC = (HIGH + LOW + 1) / 2
      IF(MIDC.EQ.ITEST) GO TO 360
      ITEST = MIDC
      IF(COMPC(IVALC,JTABLE(1,MIDC))) 357, 359, 358
357   HIGH = MIDC
      GO TO 356
358   LOW = MIDC
      GO TO 356
359   I = MAXC - MIDC + 1
      RTABLE(I) = IFREQ
      GO TO 351
360   MIDC = MINC
      IF(COMPC(IVALC,JTABLE(1,MIDC))) 917,359,917
C
C   COMPILE COLUMN STATISTICS AND HEADINGS
C
361   CONTINUE
      IF(NCOL.EQ.0) GO TO 926
      IF(NROW.EQ.0) GO TO 927
      NCOLM = NCOL
      DO 363 I = 1, NCOL
      IN = MAXC - I + 1
      J = JTABLE(4,IN)
      ITABLE(2,I) = FLD(18,18,J)
      CMEAN = 0.0
      CSD = 0.0
      J = FLD( 0,18,J)
      IVCT = IVCT + J
      V = J
      RTABLE(I) = 0
      X = CTABLE(2,IN)
      COLTOT = COLTOT + X
      X2 = CTABLE(3,IN)
      CSQTOT = CSQTOT + X2
      IF(J.EQ.0) GO TO 362
      CMEAN = X/V
      CSD = SQRT(ABS(X2/V - CMEAN*CMEAN))
362   CT(1,I) = V
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      CT(3,I) = CMEAN
      CT(4,I) = CSD
C
C  CONSTRUCT COLUMN HEADINGS
C
      INDEX = JTABLE(I,IN)
      GO TO IPASS, (501,502)
501   ITH(I) = BLK
      IBH(I) = INDEX
      GO TO 363
502   IF(INDEX.LE.INC) GO TO JPASS, (504,505,506)
      NCOLH = NCOL - 1
      IF(ISW.EQ.1) GO TO 917
      ISW = IC3 - 1
      GO TO 363
504   IBH(I) = IBC(INDEX)
      ITH(I) = ITC(INDEX)
      GO TO 363
505   X = INDEX
      BH(I) = BOTCOL + (X - 1.0) * COLINT
      TH(I) = BOTCOL + X * COLINT
      IF( TH(I).GE.TOPCOL) TH(I) = TOPCOL + 0.001
      TH(I) = TH(I) - 0.001
      GO TO 363
506   IBH(I) = MINCOL + (INDEX - 1) * INTCOL
      ITH(I) = MINCOL + INDEX * INTCOL - 1
      IF(ITH(I).GT.MAXCOL) ITH(I) = MAXCOL
363   CONTINUE
C  END LOOP
C
      ITV = -0
      LXSW = 0
      IF(IR3.EQ.2.OR.IC3.EQ.2) LXSW = 1
      IF(KODER.LE.1.OR.KODEC.LE.1) LXSW = 1
      IF(LXSW.EQ.1) GO TO 367
      DO 366 I = 1, NCOLH
366   ITV = ITV + RTABLE(I)
367   CONTINUE
      T = ITV
      IF(LXSW .EQ. 1) T = IGT
      TX = IVRT
      IF(LXSW.NE.1) TX = ITV
C
      IF(KODEC.GT.1) GO TO 503
      IF(KODEC.EQ.0) GO TO 507
      CALL SHIFT(ICCHAR,NCOLH,ITH)
507   CALL SHIFT(ICCHAR,NCOLH,IBH)
503   IF(KODER.NE.1) GO TO 508
      CALL SHIFT(IRCHAR,NROW,IBR)
      CALL SHIFT(IRCHAR,NROW,ITR)
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 508    NCOL = NCOLM + ISW
        LEF = 1
        ITITLE(LEF) = IOTHER
        LEF = LEF + ISW
        ITITLE(LEF) = IVALD
        LEF = LEF + 1
        ITITLE(LEF) = ITOTAL
        LEF = LEF + 1
        ITITLE(LEF)=MEAN
        LEF = LEF + 1
        ITITLE(LEF) = ISD
        NSTATM = LEF
        IF(IR1.EQ.1) NSTATM = NSTATM - 2
C
C
        CTM = 0.0
        CTSD = 0.0
        IF(IVCT.EQ.0) GO TO 364
        CVT = IVCT
        CTM = COLTOT / CVT
        CTSD = SQRT(ABS(CSQTOT/CVT - CTM*CTM))
 364    RTM = 0.0
        RTSD = 0.0
        IF(IVRT.EQ.0) GO TO 365
        RVT = IVRT
        RTM = ROWTOT / RVT
        RTSD = SQRT(ABS(RSQTOT/RVT - RTM*RTM))
 365    WRITE(LOU) LIMIT,ITV,IVCT,RTM,RTSD,(RTABLE(I),I=1,NCOL)
        NROW = NROW + 1
        IXT = NCOL + 1
        CT(3,IXT) = CTM
        CT(4,IXT) = CTSD
C
C
        REWIND LIN
        REWIND LOU
        N = LOU
        LOU = LIN
        LIN = N
        NPASS = (NCOL + NSTAT + 9) / 10
        IF(KODER.EQ.0) INR = LIMIT
        KX = KODER/2 + 1
C
C
C   BEGIN A LOGICAL PAGE LOOP
C
C
 515    CONTINUE
C
        IF(LSW.EQ.2) GO TO 900
```

LINE #   TEXT...

```
C
      IF(NXROW.GE.NROW) ASSIGN 750 TO LPASS
      NX = NXROW
C
      IFIRST = 1
      FIRST = 1
      LFC = 0
      LLC = 0
      KOUNT = 0
      LEF = 0
      LEFT = 0
      LPAGE = LPAGE + 1
C
C  BEGIN A PHYSICAL PAGE LOOP
C
514   CONTINUE
C
      KOUNT = KOUNT + 1
C
C  BEGIN A NEW LOGICAL PAGE IF REQUIRED PASSES ARE DONE
C
      IF(KOUNT.GT.NPASS) GO TO 515
C
C  IF THIS IS A NEW LOGICAL PAGE, DO NOT REWIND
      IF(KOUNT.EQ.1) GO TO 517
      REWIND LIN
      IF(NX.EQ.0) GO TO 517
      DO 516 I = 1, NX
516   READ(LIN)
C
C
517   IPAGE = IPAGE + 1
C
      WRITE(6,653) ID1, ID2, IPAGE
      LINE = 50
      IF(LPAGE.NE.1) GO TO 509
      IF(NHEAD.LE.0) GO TO 509
      WRITE(6,804) ((IHEAD(I,J),I=1,14),J=1,NHEAD)
      WRITE(6,666)
      LINE = LINE - (NHEAD + 1)
509   WRITE(6,648) IROW ,(LABEL(I,1),I=1,14)
      WRITE(6,649) ICOL ,(LABEL(I,2),I=1,14)
      WRITE(6,649) ISAMP,(LABEL(I,3),I=1,14)
      IF(IDS.EQ.IPLUS) GO TO 759
      WRITE(6,673) INGRP
759   WRITE(6,666)
C
C  DETERMINE COLUMN PRINT LIMITS FOR THIS PHYSICAL PAGE
C
      LFC = LLC + 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      LLC = LLC + 10
      FIRST = 1
      KTYLE = 1
      KTYPE = 1
      IF(LLC.LE.NCOLM) GO TO 511
      KTYLE = 2
      IF(LLC.LE.NCOL) GO TO 511
      LY = LLC - NCOL
      LX = 10 - LY
      IF(LX.GE.1) GO TO 510
      FIRST = 1 - LX
      IFIRST = FIRST + ISW
      IF(FIRST.GT.NSTAT) GO TO 917
      KTYPE = 3
      KTYLE = 3
      GO TO 511
510   KTYPE = 2
      KTYLE = 4
      LXH = LX + 1 - ISW
      LZ = LY
      LZH = LY + ISW
      IF(LY.LT.NSTAT) GO TO 511
      LZ = NSTAT
      LZH = NSTATH
511   CONTINUE
      GO TO (657,658,659), KY
657   JWORD = '(5XA6)'
      GO TO 656
658   JWORD = 'I11'
      GO TO 656
659   JWORD = 'F11.3'
656   ENCODE(8000,MATA) JWORD
8000  FORMAT('(12X,A4,A3,10',A6,')')
      GO TO (512,513,519,520), KTYLE
512   WRITE(6,MATA) IDC,DOT, (BH(I),I=LFC,LLC)
      WRITE(6,MATA) IDR,BLK, (TH(I),I=LFC,LLC)
      WRITE(6,661)
      GO TO LPASS, (518,750)
513   WRITE(6,MATA) IDC,DOT, (BH(I),I=LFC,NCOLM)
      WRITE(6,MATA) IDR,BLK, (TH(I),I=LFC,NCOLM)
      WRITE(6,660) DOT, (BLK,I=1,10), IOTHER
      WRITE(6,660) DOT
      GO TO LPASS, (518,750)
519   WRITE(6,MATA) IDC,DOT
      WRITE(6,MATA) IDR,BLK
      WRITE(6,660) DOT, BLK, (ITITLE(I),I=IFIRSY,NSTATH)
      WRITE(6,660) DOT
      GO TO LPASS, (518,750)
520   WRITE(6,MATA) IDC,DOT, (BH(I),I=LFC,NCOLM)
      WRITE(6,MATA) IDR,BLK, (TH(I),I=LFC,NCOLM)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      WRITE(6,660) DOT,(BLK,I=1,LXH), (ITITLE(I),I=1,LZH)
      WRITE(6,660) DOT
      GO TO LPASS, (518,750)
C
C  BEGIN THE 'PRINT A ROW' LOOP
C
518   LINE = LINE - NLR
C
C  IF NOT ENOUGH LINES FOR NEXT ROW, BEGIN NEW PHYSICAL PAGE AT 514
      IF(LINE.LE.0) GO TO 514
C
C  READ NEXT ROW FROM DISK
722   READ(LIN) INDEX, (ISTAT(I),I=1,4), (RTABLE(I),I=1,NCOL)
C  IF THIS IS A NEW ROW, INCREMENT NXROW
      IF(KOUNT.EQ.1) NXROW = NXROW + 1
C
C  DECODE THE ROW ID
C
      IF(INDEX.EQ.LIMIT) GO TO 727
      IF(INDEX.GT.INR.AND.KODER.NE.0) GO TO 726
C
      GO TO IXROW, (723,724,725,728)
C
723   IVALC = INDEX
      GO TO (776,777,778), KTYPE
C
724   IVALR = IBR(INDEX)
      IVALC = ITR(INDEX)
      GO TO (730,731,732), KTYPE
C
725   X = INDEX
      PAIR(1) = BOTROW + (X - 1.0) * ROWINT
      Y = BOTROW + X * ROWINT
      IF(Y.GE.TOPROW) Y = TOPROW + 0.001
      PAIR(2) = Y - 0.001
      GO TO (730,731,732), KTYPE
726   IVALC = IOTHER
      IF(INDEX.EQ.INVALR) GO TO 722
      GO TO (776,777,778), KTYPE
727   IVALC = IVALD
      GO TO (776,777,778), KTYPE
728   IVALR = MINROW + (INDEX - 1) * INTROW
      IVALC = MINROW + INDEX * INTROW - 1
      IF(IVALC.GT.MAXROW) IVALC = MAXROW
      GO TO (730,731,732), KTYPE
C
C  UNIVERSAL 'PRINT A ROW' SECTION (FORMATS = COMPUTED GO TO STATEMENTS)
C
730   CALL FORM2X(KX,0,0,MATF)
      WRITE(6,MATF) PAIR, (RTABLE(I),I=LFC,LLC)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      GO TO JUMRP, (733,740)
C
731   CALL FORM2X(KX,LY,0,MATE)
      WRITE(6,MATE) PAIR, (RTABLE(I),I=LFC,NCOL), (ISTAT(I),I=1,LZ)
      GO TO JUMRP, (733,740)
C
732   CALL FORM2X(KX,0,FIRST,MATC)
      WRITE(6,MATC) PAIR, (ISTAT(I),I=FIRST,NSTAT)
      GO TO JUMRP, (733,740)
C
776   WRITE(6,8681) IVALC,(RTABLE(I),I=LFC,LLC)
      GO TO JUMRP, (733,740)
C
777   CALL FORM2X(0,LY,0,MATD)
      WRITE(6,MATD) IVALC,(RTABLE(I),I=LFC,NCOL),(ISTAT(I),I=1,LZ)
      GO TO JUMRP, (733,740)
C
778   CALL FORM2X(0,0,FIRST,MATB)
      WRITE(6,MATB) IVALC,(ISTAT(I),I=FIRST,NSTAT)
      GO TO JUMRP, (733,740)
C
C  COMPUTE AND PRINT 'ROW PERCENT'
C
733   LLL = LLC
      TOT1 = ISTAT(1)
      GO TO (735,734,738), KTYPE
734   LLL = NCOL
735   IX = 0
      TOT = ISTAT(1)
      IF(LXSW.EQ.1) TOT = ISTAT(2)
      DO 736 I = LFC, LLL
      IX = IX + 1
      PCT(IX) = 0.0
      IF(TOT.EQ.0.0) GO TO 736
      CELL = RTABLE(I)
      PCT(IX) = CELL/TOT * 100.0
736   CONTINUE
      IF(KTYPE.EQ.1) GO TO 737
      IX = IX + 1
      PCT(IX) = 0.0
      IF(TOT.EQ.0.0) GO TO 737
      PCT(IX) = TOT1/TOT * 100.0
737   WRITE(6,634) IROW, (PCT(I),I=1,IX)
      GO TO JUMCP, (741,748)
738   IF(FIRST.EQ.1) GO TO 739
      WRITE(6,634) IROW
      GO TO JUMCP, (741,748)
739   X = 100.0
      IF(LXSW.EQ.1) X = TOT1 / TOT * 100.0
      WRITE(6,634) IROW, X
```

LINE #   TEXT...

```
740   GO TO JUMCP, (741,748)
C
C  COMPUTE AND PRINT 'COL PERCENT'
C
741   LLL = LLC
      GO TO (743,742,746), KTYPE
742   LLL = NCOL
743   IX = 0
      DO 744 I=LFC,LLL
      IX = IX + 1
      CELL = RTABLE(I)
      TOT = CT(1,I)
      IF(LXSW.EQ.1) TOT = ITABLE(2,I)
      PCT(IX) = 0.0
      IF(TOT.EQ.0.0) GO TO 744
      PCT(IX) = (CELL/TOT) * 100.0
744   CONTINUE
      IF(KTYPE.EQ.1) GO TO 745
      IX = IX + 1
      TT = ISTAT(1)
      PCT(IX) = 0.0
      IF(T.EQ.0.0) GO TO 745
      PCT(IX) = (TT/TX) * 100.0
745   WRITE(6,634) ICOL, (PCT(I),I=1,IX)
      GO TO 748
746   IF(FIRST.EQ.1) GO TO 747
      WRITE(6,634) ICOL
      GO TO 748
747   TT = ISTAT(1)
      P = (TT/TX) * 100.0
      IF(T.EQ.0.0) P = 0.0
      WRITE(6,634) ICOL, P
C
748   GO TO JUMTP,(775,760)
C
C  COMPUTE AND PRINT 'TOT PERCENT'
C
760   LLL = LLC
      GO TO (762,761,765), KTYPE
761   LLL = NCOL
762   IX = 0
      DO 763 I = LFC, LLL
      IX = IX + 1
      CELL = RTABLE(I)
      PCT(IX) = 0.0
      IF(T.EQ.0.0) GO TO 763
      PCT(IX) = (CELL/T) * 100.0
763   CONTINUE
      IF(KTYPE.EQ.1) GO TO 764
      IX = IX + 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      PCT(IX) = 0.0
      IF(T.EQ.0.0) GO TO 764
      TT = ISTAT(1)
      PCT(IX) = (TT/T) * 100.0
764   WRITE(6,634) JTOTAL, (PCT(I),I=1,IX)
      GO TO 775
765   IF(FIRST.EQ.1) GO TO 766
      WRITE(6,634) JTOTAL
      GO TO 775
766   TT = ISTAT(1)
      P = (TT/T) * 100.0
      IF(T.EQ.0.0) P = 0.0
      WRITE(6,634) JTOTAL, P
775   IF(INDEX.NE.LIMIT) GO TO 518
      LSW = 1
      ISAVE = 1
      LINE = LINE - 2
      IF(LINE.LE.0) GO TO 514
C  ESTABLISH THE TOTALS FOR THE COLUMNS
      ISAVE = 2
       IVALC =ITOTAL
      GO TO (749,751,752), KTYPE
749   WRITE(6,8681) IVALC,(CT(2,I),I=LFC,LLC)
      GO TO 754
751   IF(LX.GT.1) GO TO 767
      WRITE(6,8681) IVALC, (CT(2,I),I=LFC,NCOL), IVRT
      GO TO 754
767   WRITE(6,8681) IVALC, (CT(2,I),I=LFC,NCOL), IVRT,IGT
      GO TO 754
752   IF(FIRST.EQ.1) GO TO 753
      IF(FIRST.EQ.2) GO TO 768
      WRITE(6,8681) IVALC
      GO TO 754
768   WRITE(6,8681) IVALC, IGT
      GO TO 754
753   WRITE(6,8681) IVALC, IVRT, IGT
754   IF(IC1.EQ.1) GO TO 757
      LINE = LINE - 2
      IF(LINE.LT.0) GO TO 514
      WRITE(6,666)
      IF(KTYPE.EQ.3) GO TO 755
      LLL = LLC
      IF(KTYPE.EQ.2) LLL = IXT
      WRITE(6,662) ICOL,MEAN, (CT(3,I),I=LFC,LLL)
      WRITE(6,662) ICOL,ISD,  (CT(4,I),I=LFC,LLL)
      GO TO 757
755   IF(FIRST.EQ.1) GO TO 756
      WRITE(6,662) ICOL, MEAN
      WRITE(6,662) ICOL, ISD
      GO TO 900
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
756   WRITE(6,662) ICOL, MEAN, CTM
      WRITE(6,662) ICOL, ISD, CTSD
      GO TO 900
757   LSW = 2
      LINE = -1
      GO TO 514
750   IF(ISAVE.EQ.2) GO TO 754
      INDEX = LIMIT
      GO TO 748
C
C
C
634   FORMAT(8X,A3,' PERCENT',10F11.2)
C
C
C
C
648   FORMAT(1X,A3,'........',13A6,A2)
649   FORMAT(1X,A6,'....',13A6,A2)
C
652   FORMAT(9X,A3,' PERCENT',10(4X,F7.2))
653   FORMAT('12-WAY DISTRIBUTION ON TAPE-ID ',A6,A2,80X,'PAGE',I6,/)
C
C
660   FORMAT(13X,A1,A5,10(5X,A6))
661   FORMAT(13X,'.',/,13X,'.')
662   FORMAT(7X,2A6,10F11.4)
C
663   FORMAT('0  SUMMARY')
664   FORMAT('0 TOTAL NUMBER OF CASES ON TAPE',I11)
665   FORMAT('  NUMBER OF CASES IN THE SAMPLE',I11)
666   FORMAT(1X)
668   FORMAT('  STATISTICS FOR THIS SAMPLE')
669   FORMAT(11X,'STANDARD DEVIATION',3X,F10.4)
670   FORMAT(11X,'VALID',A2,I24,/,11X,'MEAN',17X,F10.4)
671   FORMAT(' * (VALID = ALL NUMERIC DATA, REGARDLESS OF RANGE.)')
672   FORMAT(' ** (VALID = ONLY IN RANGE NUMERIC DATA.)')
673   FORMAT(13X,'CASE SELECTED IF ABOVE VARIABLE = ',A1,'.')
C
C  PRINT ROWS WITH SPECIAL 'A' FORMAT
C
8681   FORMAT(/13X,A6,10I11)
C
801   FORMAT(A6,1X,A6,A2,1X,I1,1X,A4,1X,I1)
802   FORMAT(13A6,A2)
803   FORMAT(A6,1X,A1,A3,1X,3(A6,1X),3I1,3X,3F10.0)
804   FORMAT(1X,13A6,A2)
806   FORMAT('1 REQUESTED TAPE-ID = ',A6,A2,'.  TAPE MOUNTED = ',A6,A2)
C
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

DIST2X TWO WAY DISTRIBUTION PROGRAM

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 900    IPAGE = IPAGE + 1
        WRITE(6,653) ID1, ID2, IPAGE
        WRITE(6,663)
        WRITE(6,666)
        WRITE(6,804) ((IHEAD(I,J),I=1,14),J=1,NHEAD)
        WRITE(6,666)
        WRITE(6,649) ISAMP,(LABEL(I,3),I=1,14)
        IF(IDS.EQ.IPLUS) GO TO 758
        WRITE(6,673) INGRP
 758    CONTINUE
        WRITE(6,664) NCASE
        WRITE(6,665) IUSED
        WRITE(6,666)
        WRITE(6,668)
        WRITE(6,666)
        WRITE(6,648) IROW, (LABEL(I,1),I=1,14)
        WRITE(6,670) KKC, IVCT, CTM
        WRITE(6,669) CTSD
        WRITE(6,666)
        WRITE(6,649) ICOL, (LABEL(I,2),I=1,14)
        WRITE(6,670) KKR, IVRT, RTM
        WRITE(6,669) RTSD
        WRITE(6,666)
        WRITE(6,671)
        WRITE(6,672)
        IF(IXX.NE.1) GO TO 3
        STOP
 901    CALL ERRMSG(5,'** INCORRECT PROGRAM ID. *** ')
        IF(RUN.EQ.NOGO) GO TO 700
        GO TO 913
 902    CALL ERRMSG(8,'** PROGRAM IS LIMITED TO 3 HEADING CARDS. ***  ')
        IEXTRA = NHEAD - 3
        NHEAD = 3
        DO 706 J = 1, 3
        READ(5,802) (IHEAD(I,J),I=1,14)
        CALL PRINTC
 706    CONTINUE
        DO 707 I = 1, IEXTRA
        READ(5,802) JEND
        CALL PRINTC
        CALL ERRMSG(3,'** CARD IGNORED. **')
        IF(JEND.EQ.IEND) GO TO 3
 707    CONTINUE
        GO TO 2
 903    CALL ERRMSG(7,'** ROW CARD IS MISSING OR MISPUNCHED. ***')
        IF(RUN.EQ.NOGO) GO TO 702
        GO TO 913
 904    CALL ERRMSG(8,'** COLUMN CARD IS MISSING OR MISPUNCHED. ***   ')
        IF(RUN.EQ.NOGO) GO TO 703
        GO TO 913
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 905   CALL ERRMSG(8,'** NUMBER OF INTERVALS EXCEED PROGRAM LIMITS ***')
       IF(RUN.EQ.NOGO) GO TO 709
       GO TO 913
 906   CALL ERRMSG(7,'** NON-NUMERIC DATA IN ABOVE FIELD. ***  ')
       IF(RUN.EQ.NOGO) GO TO 10
       GO TO 913
 908   CALL ERRMSG(7,'** MISSING END CARD - END ASSUMED. ***   ')
       IXX = 1
       GO TO 704
 909   CALL ERRMSG(7,'** SAMPLE VARIABLE NOT IN DICTIONARY. ***' )
       GO TO 913
 910   CALL ERRMSG(7,'** COLUMN VARIABLE NOT IN DICTIONARY. ***')
       GO TO 913
 911   CALL ERRMSG(7,'** ROW VARIABLE NOT IN DICTIONARY. ***   ')
       GO TO 913
 912   CALL ERRMSG(8,'** NOGO SPECIFIED - EXECUTION SUSPENDED. ***   ')
C
 913   CONTINUE
       IF(IXX.EQ.1) GO TO 999
       IF(JSW.EQ.1) GO TO 3
       CALL ERRMSG(8,'** PROGRAM ABORTS - FOLLOWING CARDS IGNORED. **')
 914   READ(5,802,END=999) JEND
       CALL PRINTC
       IF(JEND.EQ.IEND) GO TO 3
       GO TO 914
 915   CONTINUE
       CALL ERRMSG(8,'** DUPLICATE ROW CARD - FATAL ERROR. ***       ')
       IF(RUN.NE.NOGO) GO TO 913
       GO TO 714
 916   CONTINUE
       CALL ERRMSG(8,'** DUPLICATE COLUMN CARD - FATAL ERROR. ***    ')
       IF(RUN.NE.NOGO) GO TO 913
       GO TO 715
 917   CALL ERRMSG(4,'** INTERNAL ERROR. *** ')
       GO TO 913
 918   CALL ERRMSG(8,'** UNEXPECTED BLANK CONTROL WORD - IGNORED. ***')
       GO TO 10
 919   CALL ERRMSG(8,'** UNRECOGNIZABLE CONTROL WORD - IGNORED. ***  ')
       IF(KODE.EQ.IDPRGM) GO TO 913
       GO TO 10
C
 920   CALL ERRMSG(8,'** INVALID FORMAT IN DICTIONARY FOR ABOVE ***')
       GO TO 913
C
 921   WRITE(6,806) ID1, ID2, IDT1, IDT2
       GO TO 913
 923   CALL ERRMSG(8,'** DIST2X CANNOT USE MULTIPLE WORD VARIABLES.  ')
       GO TO 913
 924   CALL ERRMSG(8,'** INVALID VARIABLE ID (MUST BE C OR V). ***   ')
       IF(RUN.EQ.NOGO) GO TO 10
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      GO TO 913
925   CALL ERRMSG(8,'** VARIABLE ID IS NOT A RIGHT-ADJUSTED NUMERIC.*')
      IF(RUN.EQ.NO;0) GO TO 10
      GO TO 913
926   CALL ERRMSG(7,'** NO VALID VALUES WITHIN COLUMN LIMITS')
      CALL ERRMSG(4,'** REPORT IS BYPASSED')
      GO TO 913
927   CALL ERRMSG(6,'** NO VALID VALUES WITHIN ROW LIMITS')
      CALL ERRMSG(4,'** REPORT IS BYPASSED')
      GO TO 913
999   CALL ERRMSG(4,'** END OF DIST2X ***   ')
      STOP
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      INTEGER FUNCTION COMPA(LAV1,LAV2)
C  ...PART OF THE DIST2X SYSTEM
      DATA LIMIT /03777777777777/
      IF(LAV1.EQ.LIMIT) GO TO 8
      IF(FLD( 0,18,LAV1).GT.FLD( 0,18,LAV2)) GO TO 8
      IF(FLD( 0,18,LAV1).LT.FLD( 0,18,LAV2)) GO TO 9
      IF(FLD(18,18,LAV1).GT.FLD(18,18,LAV2)) GO TO 8
      IF(FLD(18,18,LAV1).LT.FLD(18,18,LAV2)) GO TO 9
      COMPA = 0
      RETURN
    8 COMPA = -1
      RETURN
    9 COMPA = 1
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      INTEGER FUNCTION COMPC(LAVC,LAVD)
C  ...PART OF THE DIST2X SYSTEM
      COMMON /COMODE/COMODE(2)
      INTEGER COMPN,COMPA,COMODE
      JUMP = COMODE(2) + 1
      GO TO (6,7),JUMP
 6    COMPC = COMPN(LAVC,LAVD)
      RETURN
 7    COMPC = COMPA(LAVC,LAVD)
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      INTEGER FUNCTION COMPN(LAV1,LAV2)
C  ...PART OF THE DIST2X SYSTEM
      COMPN = 0
      IF(LAV1.GT.LAV2) COMPN = -1
      IF(LAV1.LT.LAV2) COMPN = 1
      RETURN
      END
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      INTEGER FUNCTION COMPRC(LAVA,LAVB)
C  ...PART OF THE DIST2X SYSTEM
      DIMENSION LAVA(2),LAVB(2)
      INTEGER COMPN,COMPA,COMODE
      COMMON /COMODE/COMODE(2)
      DO 4 I = 1,2
      JUMP = COMODE(I) + 1
      GO TO (1,2),JUMP
1     COMPRC = COMPN(LAVA(I),LAVB(I))
      GO TO 3
2     COMPRC = COMPA(LAVA(I),LAVB(I))
3     IF(COMPRC) 5,4,5
4     CONTINUE
5     RETURN
      END
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      SUBROUTINE DISTXI(MAXN,IBOT,ITOP,IFIND,ICOMP,ISUB)
      DIMENSION IBOT(MAXN), ITOP(MAXN)
      MAX = MAXN
      MIN = 1
      ISAME = MAX + 1
 1    ISUB = (MAX + MIN + 1) / 2
      IF(ISUB.EQ.ISAME) GO TO 5
      ISAME = ISUB
      IF(ICOMP(IFIND,IBOT(ISUB))) 2,6,4
 2    IF(ICOMP(IFIND,ITOP(ISUB))) 3,6,6
 3    MIN = ISUB
      GO TO 1
 4    MAX = ISUB
      GO TO 1
 5    IF(MAX.NE.2) GO TO 7
      MAX = 1
      GO TO 1
 7    ISUB = 0
 6    RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE FORM2X(KX,LY,FIRST,MAT)
C  ...PART OF THE DIST2X SYSTEM
C  A ZERO VALUE FOR KX, LY, OR FIRST INDICATES NON-USE.
      INTEGER FIRST
      DIMENSION MAT(1)
      IFLAG = 1
      IF(KX.NE.0) IFLAG = IFLAG + 1
      IF(LY.NE.0) IFLAG = IFLAG + 2
      IF(FIRST.NE.0) IFLAG = IFLAG + 4
      GO TO (9,1,2,4,3,5,9,9),IFLAG
1     GO TO (10,11,12),KX
10    MWORDA = 'A'
      NWORDA = '10111'
      IWORDA = ' '
      GO TO 15
11    MWORDA = 'I'
      NWORDA = '10111'
      IWORDA = ' '
      GO TO 15
12    MWORDA = '10111'
      NWORDA = ' '
      ENCODE(8070,MAT) MWORDA,NWORDA
      RETURN
15    ENCODE(8060,MAT) MWORDA,MWORDA,NWORDA,IWORDA
      RETURN
2     GO TO (281,281,282,283,284,285,286,287,288),LY
281   NWORD = '10'
      IWORD = ' '
      GO TO 2741
282   NWORD = ' 9'
      IWORD = ',F11.4'
      GO TO 2741
283   NWORD = '8111'
      GO TO 274
284   NWORD = '7111'
      GO TO 274
285   NWORD = '6111'
      GO TO 274
286   NWORD = '5111'
      GO TO 274
287   NWORD = '4111'
      GO TO 274
288   NWORD = '3111'
274   ENCODE(8010,MAT) NWORD
      RETURN
2741  ENCODE(8050,MAT) NWORD,IWORD
8050  FORMAT('(/13X,A6,',A2,'111',A6,')')
      RETURN
3     GO TO (389,390,391,391),FIRST
389   KWORD = '2111'
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      GO TO 375
390   KWORD = '1111'
      GO TO 375
391   KWORD = ' '
375   ENCODE(8010,MAT) KWORD
8010  FORMAT('(/13X,A6,',A4,',2F11.4)')
      RETURN
4     GO TO (439,440,441),KX
439   GO TO (401,401,402,403,404,405,406,407,408),LY
440   GO TO (411,411,412,413,414,415,416,417,418),LY
441   GO TO (421,421,422,423,424,425,426,427,428),LY
401   JWORDA = 'A'
      KWORDA = '10111'
      LWORDA = ' '
      GO TO 4343
402   JWORDA = 'A'
      KWORDA = '9111,'
      LWORDA = 'F11.4'
      GO TO 4343
403   JWORDA = 'A'
      KWORDA = '8111'
      GO TO 4342
404   JWORDA = 'A'
      KWORDA = '7111'
      GO TO 4342
405   JWORDA = 'A'
      KWORDA = '6111'
      GO TO 4342
406   JWORDA = 'A'
      KWORDA = '5111'
      GO TO 4342
407   JWORDA = 'A'
      KWORDA = '4111'
      GO TO 4342
408   JWORDA = 'A'
      KWORDA = '3111'
      GO TO 4342
411   JWORDA = 'I'
      KWORDA = '10111'
      LWORDA = ' '
      GO TO 4343
412   JWORDA = 'I'
      KWORDA = '9111,'
      LWORDA = 'F11.4'
      GO TO 4343
413   JWORDA = 'I'
      KWORDA = '8111'
      GO TO 4342
414   JWORDA = 'I'
      KWORDA = '7111'
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
         GO TO 4342
 415     JWORDA = 'I'
         KWORDA = '6I11'
         GO TO 4342
 416     JWORDA = 'I'
         KWORDA = '5I11'
         GO TO 4342
 417     JWORDA = 'I'
         KWORDA = '4I11'
         GO TO 4342
 418     JWORDA = 'I'
         KWORDA = '3I11'
         GO TO 4342
 421     JWORDA = '10I11'
         KWORDA = ' '
         GO TO 4341
 422     JWORDA = '9I11,'
         KWORDA = 'F11.4'
         GO TO 4341
 423     JWORDA = '8I11'
         GO TO 4340
 424     JWORDA = '7I11'
         GO TO 4340
 425     JWORDA = '6I11'
         GO TO 4340
 426     JWORDA = '5I11'
         GO TO 4340
 427     JWORDA = '4I11'
         GO TO 4340
 428     JWORDA = '3I11'
 4340    ENCODE(8020,MAT) JWORDA
         RETURN
 4341    ENCODE(8070,MAT) JWORDA,KWORDA
 8070    FORMAT('(2(/1X,F15.3),3X,',2A5,')')
         RETURN
 4342    ENCODE(8030,MAT) JWORDA,JWORDA,KWORDA
         RETURN
 4343    ENCODE(8060,MAT) JWORDA,JWORDA,KWORDA,LWORDA
 8060    FORMAT('(/4X,',A1,'6,3H - ,',A1,'6,',2A5,')')
         RETURN
 5       GO TO (542,543,544),KX
 542     GO TO (509,510,531,531),FIRST
 543     GO TO (519,520,532,532),FIRST
 544     GO TO (529,530,533,533),FIRST
 509     MWORD = 'A'
         LWORD = '2I11'
         GO TO 537
 510     MWORD = 'A'
         LWORD = '1I11'
         GO TO 537
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 531   MWORD = 'A'
       LWORD = ' '
       GO TO 537
 519   MWORD = 'I'
       LWORD = '2I11'
       GO TO 537
 520   MWORD = 'I'
       LWORD = 'I111'
       GO TO 537
 532   MWORD = 'I'
       LWORD = ' '
       GO TO 537
 529   LWORD = '2I11'
       GO TO 5371
 530   LWORD = 'I111'
       GO TO 5371
 533   LWORD = ' '
 5371  ENCODE(8020,MAT) LWORD
 8020  FORMAT('(2(/1X,F15.3),3X,',A4,',2F11.4)')
       RETURN
 537   ENCODE(8030,MAT) MWORD,MWORD,LWORD
 8030  FORMAT('(/4X,',A1,'6,3H - ,',A1,'6,',A4,',2F11.4)')
       RETURN
 9     WRITE(6,8009) KX,LY,FIRST
 8009  FORMAT(///' ILLEGAL COMBINATION OF PARAMETER VALUES FOR FORM2X.'
      1 ,5X,'KX=',I2,'  LY=',I2,'  FIRST=',I2/' RUN ABORTS')
       CALL ERTMAN(2)
       END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE OVRFLO(MINF,MAXF,N,LIMIT,KSW,ITABLE,LIN,LOU)
C  ...PART OF THE DIST2X SYSTEM
      DIMENSION ITABLE(5,1), JTABLE(2), TABLE(2), JK(2)
      EQUIVALENCE (JTABLE,TABLE),(JVAL,JK(1)),(KVAL,JK(2))
      INTEGER COMPRC
      MINF = MINF + 1
      N = MAXF
      READ(LIN) JVAL,KVAL,IFQ,SR,SQ
      IF(JVAL.EQ.LIMIT) KSW = 1
C
C  BEGIN DISK - CORE MERGE LOOP
207   IF(COMPRC(JVAL,ITABLE(1,N))) 209,213,208
C
208   WRITE(LOU) JVAL,KVAL,IFQ,SR,SQ
      READ(LIN)  JVAL,KVAL,IFQ,SR,SQ
      IF(JVAL.NE.LIMIT) GO TO 207
      KSW = 1
209   WRITE(LOU) (ITABLE(I,N),I=1,5)
210   N = N - 1
      IF(N.GE.MINF) GO TO 207
      IF(KSW.EQ.1) GO TO 212
211   WRITE(LOU) JVAL, KVAL, IFQ, SR, SQ
      READ(LIN)  JVAL, KVAL, IFQ, SR, SQ
      IF(JVAL.NE.LIMIT) GO TO 211
212   WRITE(LOU) (LIMIT,I=1,5)
      ENDFILE LOU
      REWIND LOU
      REWIND LIN
      N = LIN
      LIN = LOU
      LOU = N
      RETURN
213   IFQ = IFQ + ITABLE(3,N)
      JTABLE(1) = ITABLE(4,N)
      JTABLE(2) = ITABLE(5,N)
      SR = SR + TABLE(1)
      SQ = SQ + TABLE(2)
      WRITE(LOU) JVAL,KVAL,IFQ,SR,SQ
      READ(LIN)  JVAL,KVAL,IFQ,SR,SQ
      IF(JVAL.EQ.LIMIT) KSW = 1
      GO TO 210
      END
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN.

LINE #   TEXT...

```
      SUBROUTINE SHIFT(NCHAR,NWORD,IRAY)
C  ...PART OF THE DIST2X SYSTEM
      DIMENSION IRAY(1)
      IF(NCHAR) 3,3,4
4     ISHIFT = 6 - NCHAR
      IF(ISHIFT) 3,3,1
1     ENCODE(8000,MAT) ISHIFT,NCHAR
8000  FORMAT('(',I1,'XA',I1,')')
      DO 2 I = 1,NWORD
      IHOLD = IRAY(I)
2     ENCODE(MAT,IRAY(I)) IHOLD
3     RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      DIMENSION JA(50),IREL(2000),IDUTY(26),KA(1000),KB(1000),KC(1726),
     1KSUM(26),KNUM(26),KPER(26),ITASKS(1726),IDICT1(390),IDICT2(390),
     2IDICT3(390),KDICT(15,16),KTIT1(84),INDEX1(14),KTIT2(84),
     3INDEX2(14),KTIT3(84),INDEX3(14),JOUT(2), LTITLE(14),
     4IDCTA1(105),IDCTA2(105),IDCTA3(105),IDCTA4(75),IDCTB1(105),
     5IDCTB2(105),IDCTB3(105),IDCTB4(75),IDCTC1(105),IDCTC2(105),
     6IDCTC3(105),IDCTC4(75),JDUTY(26)
      DATA IDCTA1/630HV000 = PERCENT TIME SPENT IN DUTY A
     1                                  1A3V000 = PERCENT TIME SPENT
     1                                                               1A3V0
     2IN DUTY B
     300 = PERCENT TIME SPENT IN DUTY C
     4                    1A3V000 = PERCENT TIME SPENT IN DUTY D
     4                                              1A3V000 = PERCENT TIME
     5
     6SPENT IN DUTY E
     7 1A3V000 = PERCENT TIME SPENT IN DUTY F
     8                          1A3V000 = PERCENT TIME SPENT IN DUTY G
     8                                                  1A3/
     9
      DATA IDCTA2/630HV000 = PERCENT TIME SPENT IN DUTY H
     1                                  1A3V000 = PERCENT TIME SPENT
     1                                                               1A3V0
     2IN DUTY I
     300 = PERCENT TIME SPENT IN DUTY J
     4                    1A3V000 = PERCENT TIME SPENT IN DUTY K
     4                                              1A3V000 = PERCENT TIME
     5
     6SPENT IN DUTY L
     7 1A3V000 = PERCENT TIME SPENT IN DUTY M
     8                          1A3V000 = PERCENT TIME SPENT IN DUTY N
     8                                                  1A3/
     9
      DATA IDCTA3/630HV000 = PERCENT TIME SPENT IN DUTY O
     1                                  1A3V000 = PERCENT TIME SPENT
     1                                                               1A3V0
     2IN DUTY P
     300 = PERCENT TIME SPENT IN DUTY Q
     4                    1A3V000 = PERCENT TIME SPENT IN DUTY R
     4                                              1A3V000 = PERCENT TIME
     5
     6SPENT IN DUTY S
     7 1A3V000 = PERCENT TIME SPENT IN DUTY T
     8                          1A3V000 = PERCENT TIME SPENT IN DUTY U
     8                                                  1A3/
     9
      DATA IDCTA4/450HV000 = PERCENT TIME SPENT IN DUTY V
     1                                  1A3V000 = PERCENT TIME SPENT
     1                                                               1A3V0
     2IN DUTY W
     300 = PERCENT TIME SPENT IN DUTY X
     4                    1A3V000 = PERCENT TIME SPENT IN DUTY Y
     4                                              1A3V000 = PERCENT TIME
     5
     6SPENT IN DUTY Z
     7 1A3/
      DATA IDCTB1/630HV000 = NUMBER OF TASKS PERFORMED IN DUTY A
     1                                  1A3V000 = NUMBER OF TASKS PER
     1                                                               1A3V0
     2FORMED IN DUTY B
     300 = NUMBER OF TASKS PERFORMED IN DUTY C
     4                    1A3V000 = NUMBER OF TASKS PERFORMED IN DUTY D
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
                                                    1A3VOOD = NUMBER OF TAS
     SKS PERFORMED IN DUTY E
     7 1A3VOOD = NUMBER OF TASKS PERFORMED IN DUTY F
     8                                1A3VOOD = NUMBER OF TASKS PERFORMED IN DU
     9TY G                                                          1A3/
      DATA IDCT82/630HVOOD = NUMBER OF TASKS PERFORMED IN DUTY H
     1                                          1A3VOOD = NUMBER OF TASKS PER
     2FORMED IN DUTY I                                                   1A3VO
     300 = NUMBER OF TASKS PERFORMED IN DUTY J
     4                           1A3VOOD = NUMBER OF TASKS PERFORMED IN DUTY K
     5                                                   1A3VOOD = NUMBER OF TAS
     6KS PERFORMED IN DUTY L
     7 1A3VOOD = NUMBER OF TASKS PERFORMED IN DUTY M
     8                                1A3VOOD = NUMBER OF TASKS PERFORMED IN DU
     9TY N                                                          1A3/
      DATA IDCT83/630HVOOD = NUMBER OF TASKS PERFORMED IN DUTY O
     1                                          1A3VOOD = NUMBER OF TASKS PER
     2FORMED IN DUTY P                                                   1A3VO
     300 = NUMBER OF TASKS PERFORMED IN DUTY Q
     4                           1A3VOOD = NUMBER OF TASKS PERFORMED IN DUTY R
     5                                                   1A3VOOD = NUMBER OF TAS
     6KS PERFORMED IN DUTY S
     7 1A3VOOD = NUMBER OF TASKS PERFORMED IN DUTY T
     8                                1A3VOOD = NUMBER OF TASKS PERFORMED IN DU
     9TY U                                                          1A3/
      DATA IDCT84/630HVOOD = NUMBER OF TASKS PERFORMED IN DUTY V
     1                                          1A3VOOD = NUMBER OF TASKS PER
     2FORMED IN DUTY W                                                   1A3VO
     300 = NUMBER OF TASKS PERFORMED IN DUTY X
     4                           1A3VOOD = NUMBER OF TASKS PERFORMED IN DUTY Y
     5                                                   1A3VOOD = NUMBER OF TAS
     6KS PERFORMED IN DUTY Z
     7 1A3/
      DATA IDCTC1/630HVOOD = PERCENT OF TASKS PERFORMED IN DUTY A
     1                                          1A3VOOD = PERCENT OF TASKS PE
     2RFORMED IN DUTY B                                                  1A3VO
     300 = PERCENT OF TASKS PERFORMED IN DUTY C
     4                           1A3VOOD = PERCENT OF TASKS PERFORMED IN DUTY D
     5                                                   1A3VOOD = PERCENT OF TA
     6SKS PERFORMED IN DUTY E
     7 1A3VOOD = PERCENT OF TASKS PERFORMED IN DUTY F
     8                                1A3VOOD = PERCENT OF TASKS PERFORMED IN D
     9UTY G                                                         1A3/
      DATA IDCTC2/[illegible] PERCENT OF TASKS PERFORMED IN DUTY H
     1                                          1A3VOOD = PERCENT OF TASKS PE
     2RFORMED IN DUTY I                                                  1A3VO
     300 = PERCENT OF TASKS PERFORMED IN DUTY J
     4                           1A3VOOD = PERCENT OF TASKS PERFORMED IN DUTY K
     5                                                   1A3VOOD = PERCENT OF TA
     [illegible]
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      7 1A3V000 = PERCENT OF TASKS PERFORMED IN DUTY M
      8                        1A3V000 = PERCENT OF TASKS PERFORMED IN D
      9UTY N                                                  1A3/
       DATA IDCTC3/630HV000 = PERCENT OF TASKS PERFORMED IN DUTY O
      1                                      1A3V000 = PERCENT OF TASKS PE
      2RFORMED IN DUTY P                                                 1A3V0
      300 = PERCENT OF TASKS PERFORMED IN DUTY Q
      4                   1A3V000 = PERCENT OF TASKS PERFORMED IN DUTY R
      5                                          1A3V000 = PERCENT OF TA
      6SKS PERFORMED IN DUTY S
      7 1A3V000 = PERCENT OF TASKS PERFORMED IN DUTY T
      8                        1A3V000 = PERCENT OF TASKS PERFORMED IN D
      9UTY U                                                  1A3/
       DATA IDCTC4/450HV000 = PERCENT OF TASKS PERFORMED IN DUTY V
      1                                      1A3V000 = PERCENT OF TASKS PE
      2RFORMED IN DUTY W                                                 1A3V0
      300 = PERCENT OF TASKS PERFORMED IN DUTY X
      4                   1A3V000 = PERCENT OF TASKS PERFORMED IN DUTY Y
      5                                          1A3V000 = PERCENT OF TA
      6SKS PERFORMED IN DUTY Z
      7 1A3/
       DATA KNEW/6HbbbbNEW/
       DATA IEND/6HEND OF/
       DATA KDUTY/6H DTY  /
       DATA KVAR/6HV000 =/
       DATA JDUTY/156H   A      B      C      D      E      F      G      H
      1I     J      K      L      M      N      O      P      Q      R      S
      2T     U      V      W      X      Y      Z  /
       DATA LTITLE/84HVARIABLE DICTIONARY CREATED BY DUVARS
      1                                                  /
       EQUIVALENCE(IDICT1(1),IDCTA1(1)), (IDICT1(106),IDCTA2(1)),
      1(IDICT1(211),IDCTA3(1)),  (IDICT1(316),IDCTA4(1)),
      2(IDICT2(1),IDCTB1(1)),  (IDICT2(106),IDCTB2(1)),
      3(IDICT2(211),IDCTB3(1)),(IDICT2(316),IDCTB4(1)),
      4(IDICT3(1),IDCTC1(1)),   (IDICT3(106),IDCTC2(1)),
      5(IDICT3(211),IDCTC3(1)), (IDICT3(316),IDCTC4(1))
       ISHF6=2**6
       ISHF12=2**12
       ISHF18=2**18
       JOUT(1)=6
        JOUT(2)=26
        IHT=10
        IOUT=15
        ISCR=20
        CALL SETUP(10,15,26,0,0,0,20,0,0,0,0,JSEQ)
C
C      READ CONTROL CARD
       READ 5,ISID1,ISID2,IOP1,ITIT1,IBCD1,IADD1,IOP2,ITIT2,IBCD2,IADD2,
      1IOP3,ITIT3,IBCD3,IADD3
5      FORMAT(7X,A6,A2,12I2)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C     READ COMMUNICATION REGION AND CHECK STUDY ID
       READ(IMT) K,(JA(I),I=1,K)
      IF(JA(1).NE.ISID1.OR.JA(2).NE.ISID2) GO TO 440
      NDUTY=JA(9)
      NCASE=JA(3)
      NDICT=JA(4)
      NTASK=JA(12)
C
C     CHECK CONTROL CARD OPTIONS AND SET INDICATORS
      IF(IOP1.GT.1.OR.IOP2.GT.1.OR.IOP3.GT.1) GO TO 450
      IOP=IOP1+IOP2+IOP3
      IF(IOP.EQ.0) GO TO 450
      IF(ITIT1.GT.6.OR.ITIT2.GT.6.OR.ITIT3.GT.6) GO TO 460
      IF(IBCD1.GT.1.OR.IBCD2.GT.1.OR.IBCD3.GT.1) GO TO 470
      IBCD=IBCD1+IBCD2+IBCD3
      IBCD1=IBCD1+1
      IBCD2=IBCD2+1
      IBCD3=IBCD3+1
      IF(IADD1.GT.1.OR.IADD2.GT.1.OR.IADD3.GT.1) GO TO 480
      NEWTAP=IADD1+IADD2+IADD3
C
C     READ TITLE AND INDEX CARDS
      IWRDS1=ITIT1*14
      IWRDS2=ITIT2*14
      IWRDS3=ITIT3*14
      IF(IOP1.EQ.0) GO TO 101
      READ 11,(INDEX1(I),I=1,8),ISPEC1
      IF(ITIT1.EQ.0) GO TO 101
      READ 10,(KTIT1(I),I=1,IWRDS1)
101   IF(IOP2.EQ.0) GO TO 103
      READ 11,(INDEX2(I),I=1,8),ISPEC2
      IF(ITIT2.EQ.0) GO TO 103
      READ 10,(KTIT2(I),I=1,IWRDS2)
103   IF(IOP3.EQ.0) GO TO 105
      READ 11,(INDEX3(I),I=1,8),ISPEC3
      IF(ITIT3.EQ.0) GO TO 105
      READ 10,(KTIT3(I),I=1,IWRDS3)
10    FORMAT(13A6,A2)
11    FORMAT(8A6,1X,A6)
C
C     DETERMINE IF NEW VARIABLES ARE TO BE ADDED TO THE CODAP TAPE
105   IF(NEWTAP.EQ.0) GO TO 150
C
C     COMPUTE NEW NDICT AND WRITE COMMUNICATION REGION
      NREC=1
      IF(NDUTY.GT.16) NREC=2
      JA(4)=NDICT+NREC*NEWTAP
      NDICTN=JA(4)
       WRITE(IOUT) K,(JA(I),I=1,K)
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C
C      READ RELATIVE LOCATION TABLE AND TASK IN DUTY TABLE.  ADD RELATIVE
C      LOCATION FOR DUTIES AT END OF RELATIVE LOCATION TABLE AND WRITE
C      LOCATION AND TASK IN DUTY TABLE ON NEW CODAP TAPE.
        READ(IMT) LL,(IREL(I),I=1,LL),MM,(IDUTY(I),I=1,MM)
       LNTH=IREL(LL)/ISHF18
       ISTART=MOD(IREL(LL),ISHF18)
       KNEW=KNEW+LNTH+ISTART
       ITOT=NDUTY*IOP
       DO 110 I=1,ITOT
       LL=LL+1
       IREL(LL)=KNEW
 110   KNEW=KNEW+1
        WRITE(IOUT) LL,(IREL(I),I=1,LL),MM,(IDUTY(I),I=1,MM)
C
C      READ AND WRITE THE DICTIONARY
       DO 120 N=1,NDICT
        READ(IMT) K,((KDICT(I,J),I=1,15),J=1,K)
   120  WRITE(IOUT) K,((KDICT(I,J),I=1,15),J=1,K)
C
C      FIND THE LAST VARIABLE NUMBER AND NUMBER NEW VARIABLE TITLES
       DECODE(605,KDICT(1,K)) KLAST
 605   FORMAT(1X,A3)
       CALL BCDBIN(KLAST,KLAST,3,$500)
       IQUIT=NDUTY*15
       IF(IOP1.NE.1) GO TO 124
       DO 123 II=1,IQUIT,15
       KLAST=KLAST+1
        CALL BINBCD(KLAST,MLAST,3,$500,0)
   123  ENCODE(610,IDICT1(II)) MLAST
   610  FORMAT('V',A3,' =')
 124   IF(IOP2.NE.1) GO TO 126
       DO 125 II=1,IQUIT,15
       KLAST=KLAST+1
        CALL BINBCD(KLAST,MLAST,3,$500,0)
   125  ENCODE(610,IDICT2(II)) MLAST
 126   IF(IOP3.NE.1) GO TO 128
       DO 127 II=1,IQUIT,15
       KLAST=KLAST+1
        CALL BINBCD(KLAST,MLAST,3,$500,0)
   127  ENCODE(610,IDICT3(II)) MLAST
 128   CONTINUE
C
C      ADD NEW DICTIONARY ENTRIES.  EACH BLOCK CAN CONTAIN A MAX OF
C      16 DEFINITIONS
        INUM=16
        IF(NDUTY.LT.INUM) INUM=NDUTY
        IX=INUM*15
        JNUM=NDUTY-16
        J=IX+1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
              IK=NDUTY*15
              IF(IOP1.NE.1) GO TO 130
               WRITE(IOUT) INUM,(IDICT1(I),I=1,IX)
              IF(NDUTY.LE.16) GO TO 130
               WRITE(IOUT) JNUM,(IDICT1(I),I=J,IK)
      130     IF(IOP2.NE.1) GO TO 140
               WRITE(IOUT) INUM,(IDICT2(I),I=1,IX)
              IF(NDUTY.LE.16) GO TO 140
               WRITE(IOUT) JNUM,(IDICT2(I),I=J,IK)
      140     IF(IOP3.NE.1) GO TO 170
               WRITE(IOUT) INUM,(IDICT3(I),I=1,IX)
              IF(NDUTY.LE.16) GO TO 170
               WRITE(IOUT) JNUM,(IDICT3(I),I=J,IK)
              GO TO 170
    C
    C         NO NEW TAPE.  PASS TAPE TO GET TO CASE DATA
      150      READ(IHT) LL,(IREL(I),I=1,LL),MM,(IDUTY(I),I=1,MM)
              DO 160 I=1,NDICT
      160      READ(IHT) K
    C
    C         READ CASE DATA AND COMPUTE DUVARS.  COMPUTE ALL 3 OPTIONS
      170      READ(IHT) K,(KA(I),I=1,K),L,(KB(I),I=1,L),M,(KC(I),I=1,M)
              IF(KA(1).EQ.IEND) GO TO 270
              IF(NEWTAP.EQ.0) GO TO 175
              KTOTAL=ITOT+L
              IF(KTOTAL.GT.928) GO TO 490
      175     DO 180 I=1,NTASK
      180     ITASKS(I)=0
              DO 185 I=1,26
              KSUM(I)=0
              KNUM(I)=0
      185     KPER(I)=0
              DO 190 I=1,M
              ITSK = KC(I)/ISHF18
              IPCNT=MOD(KC(I),ISHF18)
      190     ITASKS(ITSK)=IPCNT
              DO 200 I=1,NDUTY
              ISTOP=MOD(IDUTY(I),ISHF18)
              ISTART=IDUTY(I)/ISHF18
              DO 195 J=ISTART,ISTOP
              IF(ITASKS(J).EQ.0) GO TO 195
              KSUM(I)=KSUM(I)+ITASKS(J)
              KNUM(I)=KNUM(I)+1
      195     CONTINUE
              IF(KNUM(I).EQ.0) GO TO 200
              KSUM(I)=(KSUM(I)+500)/1000
              KPER(I)=(KNUM(I)*1000/M+5)/10
      200     CONTINUE
    C
    C         WRITE DUVARS ON A SCRATCH UNIT
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
       WRITE(ISCR)KB(1),KA(2),(KSUM(I),I=1,NDUTY),(KNUM(I),I=1,NDUTY),
      1(KPER(I),I=1,NDUTY)
       IF(NEWTAP.EQ.0) GO TO 170
C
C      ADD DUTY VARIABLES TO NEW TAPE
       IF(IOP1.NE.1) GO TO 220
       DO 210 I=1,NDUTY
       L=L+1
  210   CALL BINBCD(KSUM(I),KB(L),3,$500,1)
 220   IF(IOP2.NE.1) GO TO 240
       DO 230 I=1,NDUTY
       L=L+1
  230   CALL BINBCD(KNUM(I),KB(L),3,$500,1)
 240   IF(IOP3.NE.1) GO TO 260
       DO 250 I=1,NDUTY
       L=L+1
  250   CALL BINBCD(KPER(I),KB(L),3,$500,1)
   260  WRITE(IOUT) K,(KA(I),I=1,K),L,(KB(I),I=1,L),M,(KC(I),I=1,M)
       GO TO 170
 270   IF(NEWTAP.EQ.0) GO TO 280
        WRITE(IOUT) K,(KA(I),I=1,K),L,(KB(I),I=1,L),M,(KC(I),I=1,M)
  280   READ(IMT) K,(KA(I),I=1,K)
        WRITE(IOUT) K,(KA(I),I=1,K)
       IF(KA(1).NE.IEND) GO TO 280
        REWIND IOUT
C
C      PRINT NEW DICTIONARY OF VARIABLES
        READ(IOUT) K
        READ(IOUT) K
       CALL PRTDIC(IOUT,6,1,LTITLE,NDICTN)
        REWIND IOUT
  290   REWIND IMT
        REWIND ISCR
C
C      PRINT OPTION 1 REPORT IF REQUESTED
 295   IF(IOP1.EQ.0) GO TO 330
       IPAGE=1
       DO 305   LL=1,IBCD1
       IOUT=JOUT(LL)
       WRITE(IOUT,17)(INDEX1(I),I=1,8),ISPEC1,IPAGE
 17    FORMAT('1',8A6,29X,A6,1X,'PAGE ',I3,//)
       IF(ITIT1.EQ.0) GO TO 300
       WRITE(IOUT,19)(KTIT1(I),I=1,IWRDS1)
 19    FORMAT(1X,13A6,A2)
       WRITE(IOUT,27)
 300   WRITE(IOUT,25)(KDUTY,I=1,NDUTY)
 25    FORMAT('  SEQ  CASE NTASK',26A4)
       WRITE(IOUT,26)(JDUTY(I),I=1,NDUTY)
 26    FORMAT('  NUM  NUM        ',26A4)
        WRITE(IOUT,27)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      27 FORMAT('0')
 305  CONTINUE
      ILINE=1+ITIT1
      IPAGE=2
      DO 320 N=1,NCASE
       READ(ISCR) KASE,NTSKS,(KSUM(I),I=1,NDUTY),(KNUM(I),I=1,NDUTY),
     1(KPER(I),I=1,NDUTY)
      DO 310   LL=1,IBCD1
      IOUT=JOUT(LL)
      WRITE(IOUT,28) N,KASE,NTSKS,(KSUM(I),I=1,NDUTY)
 28   FORMAT(1X,I4,2X,A4,2X,I3,1X,26I4)
      IF(ILINE.NE.50) GO TO 310
      WRITE(IOUT,17)(INDEX1(I),I=1,8),ISPEC1,IPAGE
      WRITE(IOUT,25)(KDUTY,I=1,NDUTY)
      WRITE(IOUT,26)(JDUTY(I),I=1,NDUTY)
      WRITE(IOUT,27)
 310  CONTINUE
      IF(ILINE.NE.50) GO TO 315
      IPAGE=IPAGE+1
      ILINE = 1
      GO TO 320
 315  ILINE=ILINE+1
 320  CONTINUE
       REWIND ISCR
C
C     PUNCH REQUEST CARD FOR OPTION 1 IF REPORT WRITTEN ON BCD TAPE
      IF(IBCD1.EQ.1) GO TO 330
       CALL REPEND(ISID1,ISPEC1,INDEX1(1),JSEQ)
C
C     PRINT OPTION 2 REPORT IF REQUESTED.
 330  IF(IOP2.EQ.0) GO TO 380
      IPAGE=1
      DO 350   LL=1,IBCD2
      IOUT=JOUT(LL)
      WRITE(IOUT,17)(INDEX2(I),I=1,8),ISPEC2,IPAGE
      IF(ITIT2.EQ.0) GO TO 340
      WRITE(IOUT,19) (KTIT2(I),I=1,IWRDS2)
      WRITE(IOUT,27)
 340  WRITE(IOUT,25)(KDUTY,I=1,NDUTY)
      WRITE(IOUT,26)(JDUTY(I),I=1,NDUTY)
      WRITE(IOUT,27)
 350  CONTINUE
      ILINE=1+ITIT2
      IPAGE=2
      DO 370 N=1,NCASE
      READ(ISCR) KASE,NTSKS,(KSUM(I),I=1,NDUTY),(KNUM(I),I=1,NDUTY),
     1(KPER(I),I=1,NDUTY)
      DO 360   LL=1,IBCD2
      IOUT=JOUT(LL)
      WRITE(IOUT,28) N,KASE,NTSKS,(KNUM(I),I=1,NDUTY)
```

LINE # TEXT...

```
      IF(ILINE.NE.50) GO TO 360
      WRITE(IOUT,17)(INDEX2(I),I=1,8),ISPEC2,IPAGE
      WRITE(IOUT,25)(KDUTY,I=1,NDUTY)
      WRITE(IOUT,26)(JDUTY(I),I=1,NDUTY)
      WRITE(IOUT,27)
  360 CONTINUE
      IF(ILINE.NE.50) GO TO 365
      IPAGE=IPAGE+1
      ILINE = 1
      GO TO 370
  365 ILINE=ILINE+1
  370 CONTINUE
       REWIND ISCR
C
C     PUNCH REQUEST CARD FOR OPTION 2 IF REPORT WRITTEN ON BCD TAPE
      IF(IBCD2.EQ.1) GO TO 380
       CALL REPEND(ISID1,ISPEC2,INDEX2(1),JSEQ)
C
C     PRINT OPTION 3 REPORT IF REQUESTED
  380  IF(IOP3.EQ.0) GO TO 1001
      IPAGE=1
      DO 400 LL=1,IBCD3
      IOUT=JOUT(LL)
      WRITE(IOUT,17)(INDEX3(I),I=1,8),ISPEC3,IPAGE
      IF(ITIT3.EQ.0) GO TO 390
      WRITE(IOUT,19)(KTIT3(I),I=1,IWRDS3)
      WRITE(IOUT,27)
  390 WRITE(IOUT,25)(KDUTY,I=1,NDUTY)
      WRITE(IOUT,26)(JDUTY(I),I=1,NDUTY)
      WRITE(IOUT,27)
  400 CONTINUE
      ILINE=1+ITIT3
      IPAGE=2
      DO 420 N=1,NCASE
       READ(ISCR) KASE,NTSKS,(KSUM(I),I=1,NDUTY),(KNUM(I),I=1,NDUTY),
     1(KPER(I),I=1,NDUTY)
      DO 410   LL=1,IBCD3
      IOUT=JOUT(LL)
      WRITE(IOUT,28) N,KASE,NTSKS,(KPER(I),I=1,NDUTY)
      IF(ILINE.NE.50) GO TO 410
      WRITE(IOUT,17)(INDEX3(I),I=1,8),ISPEC3,IPAGE
      WRITE(IOUT,25)(KDUTY,I=1,NDUTY)
      WRITE(IOUT,26)(JDUTY(I),I=1,NDUTY)
      WRITE(IOUT,27)
  410 CONTINUE
      IF(ILINE.NE.50) GO TO 415
      IPAGE=IPAGE+1
      ILINE = 1
      GO TO 420
  415 ILINE=ILINE+1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
  420   CONTINUE
          REWIND ISCR
C
C       PUNCH REQUEST CARD FOR OPTION 3 IF REPORT WRITTEN ON BCD TAPE
          IF(IBCD3.EQ.1) GO TO 1000
          CALL REPEND(ISID1,ISPEC3,INDEX3(1),JSEQ)
        GO TO 1001
  440   PRINT 37, JA(1),JA(2)
  37    FORMAT('1STUDY ID NO MATCH.  TAPE ID IS ',A6,A2)
        GO TO 1000
  450   PRINT 39
  39    FORMAT('1COLUMN 17 OR 25 OR 33 OF THE DUVARS CONTROL CARD IS IN ER
       1ROR.  RUN TERMINATED.')
        GO TO 1000
  460   PRINT 42
  42    FORMAT('1COLUMN 19 OR 27 OR 35 OF THE DUVARS CONTROL CARD IS IN ER
       1ROR.  RUN TERMINATED.')
        GO TO 1000
  470   PRINT 44
  44    FORMAT('1COLUMN 21 OR 29 OR 37 OF THE DUVARS CONTROL CARD IS IN ER
       1ROR.  RUN TERMINATED.')
        GO TO 1000
  480   PRINT 46
  46    FORMAT('1COLUMN 23 OR 31 OR 39 OF THE DUVARS CONTROL CARD IS IN ER
       1ROR.  RUN TERMINATED.')
        GO TO 1000
  490   PRINT 48, KTOTAL
  48    FORMAT('1THE MAXIMUM NUMBER OF WORDS OF COMPUTED AND BACKGROUND VA
       1RIABLES ALLOWABLE IS 928.  THIS RUN WILL CREATE ',I4,'.  RUN TERMI
       2NATED.')
        GO TO 1000
  500   PRINT 50, KDICT(1,K)
  50    FORMAT('1THE LAST VARIABLE DICTIONARY ENTRY FROM THE CODAP TAPE IS
       1 ERRONEOUS IN COLUMNS 1-6.  THEY ARE ',A6)
  1000  CALL ERTRAN(2)
  1001  WRITE(6,600)
   600  FORMAT('1')
        STOP
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

EXTRCT EXTRACT REPORTS FROM THE REPORT FILES

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      INTEGER POINT, ONE, TWO, BLANK, OPTION, TITLE, HI
      DIMENSION POINT(700), IDR(700), LOW(700), IHI(700), IMAGE(22),
     1 NAME(3), TITLE(9,700)
      COMMON MAXPGE(700)
      COMMON /INFILE/ IDPROG, NAME
      DATA ONE, TWO, BLANK /'1', '2', ' '/
      CALL SETUPS(28,15,0,0,0,0,20,0,0,0,0,JSEQ)
      CALL RESET
      READ(5,800) OPTION, ISTUDY, NCOPY, NPAGE, ISTART
      CALL PRINTC
      IF(OPTION.EQ.TWO) GO TO 995
      IF(NCOPY.EQ.BLANK) NCOPY = ONE
      CALL BCDBIN(NCOPY,NCOPY,1,$777)
      CALL BCDBIN(NPAGE,NPAGE,2,$778)
      IF(NPAGE.LE.0) NPAGE = 99
      IF(OPTION.EQ.ONE) GO TO 990
      NRQST = 1
    1 READ(5,801,END=2) (TITLE(I,NRQST),I=1,9), MAX, JSEQ
      CALL PRINTC
      CALL BCDBIN(MAX,MAXPGE(NRQST),2,$776)
      CALL BCDBIN(JSEQ,IDR(NRQST),4,$775)
      NRQST = NRQST + 1
      GO TO 1
    2 IF(NRQST.EQ.1) GO TO 774
      NRQST = NRQST - 1
      IF(NRQST.GT.700) GO TO 773
      DO 3 INDEX = 1, NRQST
      POINT(INDEX) = INDEX
    3 CONTINUE
      IF(NRQST.GT.1) CALL LISORT(NRQST,IDR,POINT)
C   DEFINE THE SCRATCH FILE
C   60 LINES/PAGE TIMES 25 PAGES/REPORT = 1500 LINES/REPORT
      NREC = NRQST*1500
      IO = 29
      DEFINE FILE IO(NREC,23,U,NEXT)
C
C   SETUP DUMMY JSEQ VALUE TO START DISK COPY LOOP
      JSEQ = -1
C   SET RECORD INDEX AND NULL REPORT SWITCH
      IREC = 0
      ISW = 1
      DO 8 INDEX = 1, NRQST
      LREC = IREC + 1500
      IF(LREC.LT.NREC) GO TO 15
      N = NRQST - INDEX + 1
      NREC = NREC + N*1000
      DEFINE FILE IO(NREC,23,U,NEXT)
   15 CONTINUE
      LOC = POINT(INDEX)
      NUM = IDR(LOC)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

```
TEXT...

         MSEQ = JSEQ - 1
         IF(MSEQ.NE.NUM) GO TO 4
         JOC = POINT(INDEX-1)
         LOW(LOC) = LOW(JOC)
         IHI(LOC) = IHI(JOC)
         GO TO 13
  4      IF(JSEQ.NE.NUM) GO TO 6
         LOW(LOC) = IREC + 1
         ASSIGN 30 TO JUMP
  5      CONTINUE
         CALL TREAD(NWORD,IMAGE,JSEQ,$7,$771)
         GO TO JUMP
  30     ASSIGN 25 TO JUMP
         IF(TITLE(1,LOC).NE.' '.AND.TITLE(2,LOC).NE.' ') GO TO 25
         TITLE(1,LOC) = IMAGE(14)
         DECODE(137,IMAGE(1)) IWORD
  137    FORMAT(1X,A5)
         ENCODE(183,TITLE(2,LOC)) IWORD, (IMAGE(I),I=2,9)
  183    FORMAT(A5,7A6,A1)
  25     CONTINUE
         IREC = IREC + 1
         WRITE(10'IREC,ERR=770) NWORD, (IMAGE(I),I=1,NWORD)
         GO TO 5
  6      CALL TREAD(NWORD,IMAGE,JSEQ,$4,$772)
         GO TO 6
  7      IHI(LOC) = IREC
  13     IF(LOW(LOC).LE.IREC) GO TO 8
         IF(ISW.EQ.1) WRITE(6,805)
         ISW = 0
         POINT(INDEX) = 0
         WRITE(6,806) (TITLE(I,LOC),I=1,9), IDR(LOC)
  8      CONTINUE
         REWIND 15
         NUM = 0
         DO 10 INDEX = 1, NRQST
         LOC = POINT(INDEX)
         IF(LOC.EQ.0) GO TO 10
         NUM = NUM + 1
         WRITE(15,810) NUM
         LO = LOW(INDEX)
         HI = IHI(INDEX)
         DO 9 IREC = LO, HI
         READ(10'IREC,ERR=769) NWORD, (IMAGE(I),I=1,NWORD)
         WRITE(15,811) (IMAGE(I),I=1,NWORD)
  9      CONTINUE
  10     CONTINUE
         NUM = NUM + 1
         WRITE(15,810) NUM
         ENDFILE 15
         REWIND 15
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      ENCODE(807,IMAGE) NAME
      CALL ERTRAN(6,IMAGE)
      ENCODE(808,IMAGE)
      CALL ERTRAN(6,IMAGE)
      ENCODE(809,IMAGE)
      CALL ERTRAN(6,IMAGE)
      NUM = 0
      DO 11 INDEX = 1, NRQST
      LOC = POINT(INDEX)
      IF(LOC.EQ.0) GO TO 11
      NUM = NUM + 1
      IPU = 1
      WRITE(IPU,812) ISTUDY, (TITLE(I,INDEX),I=1,9), NUM
   11 CONTINUE
      CALL REWIND
      CALL EDTPRT(NPAGE,NCOPY)
      GO TO 999
  769 CALL ERRMSG(6,'*** RANDOM READ ERROR OCCURRED ')
      GO TO 998
  770 CALL ERRMSG(6,'*** RANDOM WRITE ERROR OCCURRED')
      GO TO 998
  771 JSEQ = JSEQ - 1
  772 WRITE(6,804) JSEQ, NUM
      GO TO 998
  773 WRITE(6,803) NRQST
      GO TO 998
  774 CALL ERRMSG(9,'*** THE RUN EXPECTED REQUEST CARDS BUT NONE FOUND')
      GO TO 998
  775 CALL ERRMSG(7,'*** INVALID REPORT NUMBER IN CC 77-80')
      GO TO 998
  776 CALL ERRMSG(7,'*** INVALID PAGE MAX IN CC 75-76 ABOVE')
      GO TO 998
  777 CALL ERRMSG(7,'*** INVALID NUMBER OF COPIES REQUESTED')
      GO TO 998
  778 CALL ERRMSG(5,'*** INVALID PAGE MAXIMUM ABOVE')
      GO TO 998
  779 CALL ERRMSG(7,'*** INVALID STARTING PAGE NUMBER ABOVE')
      GO TO 998
  990 CONTINUE
      CALL BCDBIN(ISTART,ISTART,3,$779)
      CALL PRINTR(NPAGE,NCOPY,ISTART)
      GO TO 999
  995 CONTINUE
      CALL RPINDX
      GO TO 999
  998 CALL ERRMSG(5,'RUN HALTS DUE TO ABOVE ERROR')
      CALL MARGIN
      CALL ERTRAN(2)
  999 CALL MARGIN
      STOP
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 800  FORMAT(7X,A1,1X,A4,1X,A1,1X,A2,1X,A3)
 801  FORMAT(4X,9A6,16X,A2,A4)
 803  FORMAT('0THIS RUN EXCEEDS THE 700 REQUEST CARD LIMIT.  READ ',I5)
 804  FORMAT('0END OF FILE AFTER REPORT ',I3,'. SEARCHING FOR ',I3)
 805  FORMAT('1THE FOLLOWING REPORTS CONTAINED NO PRINT LINES AND WERE I
     1GNORED',/,' REPORT ID   REPORT TITLE',39X,'NUMBER',/)
 806  FORMAT(4X,A6,3X,8A6,3X,I6)
 807  FORMAT(6HWFREE ,3A6)
 808  FORMAT(11HWFREE 28 . )
 809  FORMAT(13HWUSE 28,15 . )
 810  FORMAT('BEGIN REPORT',A6)
 811  FORMAT(22A6)
 812  FORMAT(A4,9A6,18X,I4)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE PRINTR(NPAGES,NCOPY,ISTART)
      DIMENSION LOC(23)
      INTEGER PAGE, REPORT, ONE, TEST, TEMP, START
      DATA ONE/'1'/
      PAGE = 0
      REPORT = 0
      INDEX = 1
      START = ISTART
      ASSIGN 4 TO ICOPY
C  BEGIN REPORT PRINT LOOP
10    IPAGE = 0
      IPAGE = IPAGE + 1
1     CALL TREAD(NWORD,LOC,JSEQ,$3,$999)
      DECODE(100,LOC) TEST
      IF(TEST.NE.ONE) GO TO 2
      IPAGE = IPAGE + 1
      IF(IPAGE.GT.NPAGES) GO TO 5
      PAGE = PAGE + 1
      ENCODE(106,LOC(16)) PAGE
2     IF(PAGE.GE.START) WRITE(6,101) (LOC(I),I=1,NWORD)
      GO TO 1
3     CALL TREAD(NWORD,LOC,LSEQ,$3,$777)
      PAGE = PAGE + 1
      ENCODE(106,LOC(16)) PAGE
      IF(PAGE.GE.START) WRITE(6,101) (LOC(I),I=1,NWORD)
      GO TO ICOPY
4     REPORT = REPORT + 1
      DECODE(102,LOC) TEMP
      ENCODE(102,LOC) TEMP
      WRITE(20) JSEQ, LOC(14), (LOC(I),I=1,9), PAGE
      GO TO 10
5     CALL TREAD(NWORD,LOC,JSEQ,$3,$777)
      GO TO 5
777   REWIND 20
      DO 778 IREP = 1, REPORT
      IF(MOD(IREP,55).EQ.1) WRITE(6,103)
      READ(20) JSEQ, ID, (LOC(I),I=1,9), PAGEI
      IF(PAGE.GE.START) WRITE(6,104) JSEQ, ID, (LOC(I),I=1,9), PAGEI
778   CONTINUE
      IF(INDEX.GE.NCOPY) RETURN
      INDEX = INDEX + 1
      ASSIGN 10 TO ICOPY
      PAGE = 0
      START = 0
C     FOR TAPE FILES REWIND UNIT 28
      CALL REWIND
      GO TO 10
999   WRITE(6,105)
      GO TO 777
100   FORMAT(1A1)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
101   FORMAT(22A6)
102   FORMAT(1X,A5)
103   FORMAT('1',47X,'TABLE OF CONTENTS',//,24X,'REPORT   REPORT ID',
     16X,'REPORT TITLE',43X,'PAGE',/,24X,'NUMBER',71X,'NUMBER',/)
104   FORMAT(25X,I4,7X,A6,5X,8A6,A1,7X,I4)
105   FORMAT('1REPORT FILE IS MISSING LAST SENTINEL. RUN CONTINUES')
106   FORMAT(I3)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
        SUBROUTINE EDTPRT(NPAGE,NCOPY)
        COMMON MAXPGE(700)
        DIMENSION LOC(23)
        INTEGER PAGE, REPORT, ONE, TEST, TEMP
        DATA ONE/'1'/
        PAGE = 0
        REPORT = 0
        INDEX = 1
        ASSIGN 4 TO ICOPY
 9      NEXT = 0
        PAGE = 0
C  BEGIN REPORT PRINT LOOP
10      IPAGE = 0
        IPAGE = IPAGE + 1
 1      CALL TREAD(NWORD,LOC,JSEQ,$3,$999)
        DECODE(100,LOC) TEST
        IF(TEST.NE.ONE) GO TO 2
        IPAGE = IPAGE + 1
        IF(IPAGE.GT.NPAGES) GO TO 5
        PAGE = PAGE + 1
        ENCODE(106,LOC(16)) PAGE
 2      WRITE(6,101) (LOC(I),I=1,NWORD)
        GO TO 1
 3      CALL TREAD(NWORD,LOC,LSEQ,$3,$777)
        NEXT = NEXT + 1
        LIMIT = MAXPGE(NEXT)
        NPAGES = NPAGE
        IF(LIMIT.NE.0) NPAGES = LIMIT
        PAGE = PAGE + 1
        ENCODE(106,LOC(16)) PAGE
        WRITE(6,101) (LOC(I),I=1,NWORD)
        GO TO ICOPY
 4      REPORT = REPORT + 1
        DECODE(102,LOC) TEMP
        ENCODE(102,LOC) TEMP
        WRITE(20) JSEQ, LOC(14), (LOC(I),I=1,9), PAGE
        GO TO 10
 5      CALL TREAD(NWORD,LOC,JSEQ,$3,$777)
        GO TO 5
777     REWIND 20
        DO 778 IREP = 1, REPORT
        IF(MOD(IREP,55).EQ.1) WRITE(6,103)
        READ(20) JSEQ, ID, (LOC(I),I=1,9), PAGE
        WRITE(6,104) JSEQ, ID, (LOC(I),I=1,9), PAGE
778     CONTINUE
        IF(INDEX.GE.NCOPY) RETURN
        INDEX = INDEX + 1
        ASSIGN 10 TO ICOPY
C       FOR TAPE FILES REWIND UNIT 28
        CALL REWIND
```

PROGRAM TYPE: SUBPROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          GO TO 9
    999   WRITE(6,105)
          GO TO 777
    100   FORMAT(1A1)
    101   FORMAT(22A6)
    102   FORMAT(1X,A5)
    103   FORMAT('1',47X,'TABLE OF CONTENTS',//,24X,'REPORT   REPORT ID',
         16X,'REPORT TITLE',43X,'PAGE',/,24X,'NUMBER',71X,'NUMBER',/)
    104   FORMAT(25X,I4,7X,A6,5X,8A6,A1,7X,I4)
    105   FORMAT('1REPORT FILE IS MISSING LAST SENTINEL. RUN CONTINUES')
    106   FORMAT(I3)
          END
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE SETUPS(I1,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,JSEQ)
      DIMENSION LIST(6),  ID(3,6), IMAGE(7), NAME(3), INPUT(3)
      INTEGER BLANK, UNIT
      COMMON/INFILE/ IDPROG, INPUT
      DATA BLANK/' '/, ISCRTH/'SCRATC'/
      JSEQ = 0
      LIST(1) = I1
      LIST(2) = I2
      LIST(3) = I3
      LIST(4) = I4
      LIST(5) = I5
      LIST(6) = I6
      READ(5,800) IDPROG, ID
800   FORMAT(A6,1X,6(A6,A4,A1,1X))
      WRITE(6,112) IDPROG
      DO 99 L = 6, 1, -1
      IF(LIST(L).NE.25.AND.LIST(L).NE.26) GO TO 99
      IF(ID(1,L).EQ.' ') GO TO 99
      IF(ID(1,L).NE.'SCRATC') GO TO 80
      ENCODE(108,IMAGE) LIST(L)
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      ENCODE(106,IMAGE) L
      WRITE(6,111) (IMAGE(I),I=1,4)
      CALL GETFIL(IMAGE)
      ENCODE(105,IMAGE) LIST(L), L
      WRITE(6,111) (IMAGE(I),I=1,4)
      CALL ERTRAN(6,IMAGE)
      GO TO 99
80    CONTINUE
      UNIT = LIST(L)
           LIST(L) = 0
           CALL CYCLES(ID(1,L),UNIT,IASGD)
           IF(IASGD.EQ.1) GO TO 95
           IF(UNIT.NE.25) GO TO 90
           IOLD = 27
           CALL JDCOPY
           CALL FREE(IOLD)
           GO TO 99
90         IOLD = 28
           CALL RPCOPY(JSEQ)
           CALL FREE(IOLD)
           GO TO 99
95         IF(UNIT.NE.25) GO TO 97
           CALL JDEOF
           GO TO 99
97         CALL RPEOF(UNIT,JSEQ)
99         CONTINUE
       DO 49 L = 1, 6
       IF(LIST(L).EQ.0) GO TO 49
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      IF(ID(1,L).EQ.BLANK.AND.ID(2,L).EQ.BLANK) GO TO 16
      UNIT = LIST(L)
      IF(ID(1,L).EQ.ISCRTM) GO TO 15
      IF(UNIT.GT.28.OR.UNIT.LT.10) GO TO 777
      IF(UNIT.GT.14) GO TO 4
      IF(ID(3,L).EQ.BLANK) GO TO 1
      ENCODE(100,NAME)   (ID(I,L),I=1,3)
      GO TO 2
 1    ENCODE(101,NAME)   (ID(I,L),I=1,2)
 2    CONTINUE
      IF(UNIT.EQ.10) INPUT(1) = NAME(1)
      IF(UNIT.EQ.10) INPUT(2) = NAME(2)
      IF(UNIT.EQ.10) INPUT(3) = NAME(3)
      CALL ASGA(ID(1,L),UNIT)
      GO TO 49
 4    IF(UNIT.GT.19) GO TO 6
      CALL ASGCM(ID(1,L),UNIT)
      GO TO 49
 6    IF(UNIT.GT.24) GO TO 14
 15   ENCODE(108,IMAGE) UNIT
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      ENCODE(106,IMAGE) L
      WRITE(6,111) (IMAGE(I),I=1,4)
      CALL GETFIL(IMAGE)
      ENCODE(105,IMAGE) UNIT, L
      WRITE(6,111) (IMAGE(I),I=1,4)
          CALL ERTRAN(6,IMAGE)
          GO TO 49
 16   ENCODE(108,IMAGE) LIST(L)
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      ENCODE(115,IMAGE) LIST(L), L
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      GO TO 49
 14   CALL ASGAM(ID(1,L),UNIT)
      IF(ID(3,L).EQ.BLANK) ID(3,L) = '0'
      ENCODE(177,INPUT) (ID(I,L),I=1,3)
 177  FORMAT(A6,A4,'M(-',A1,')',)
 49   CONTINUE
      LIST(1) = J1
      LIST(2) = J2
      LIST(3) = J3
      LIST(4) = J4
      LIST(5) = J5
      DO 50 I = 1, 5
      IF(LIST(I).EQ.0) GO TO 50
      ENCODE(109,IMAGE) LIST(I)
      WRITE(6,111) (IMAGE(KK),KK=1,5)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

SETUPS FILE ASSIGNMENT ROUTINE FOR EXTRCT

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
            CALL GETFIL(IMAGE)
      50    CONTINUE
            CALL NOHEAD
            RETURN
     777    WRITE(6,110)
            CALL ERTRAN(2)
     100    FORMAT( A6,A4,2H(-,A1,4H) . )
     101    FORMAT( A6,A4,3H . )
     105    FORMAT('@USE ',I2,',SCRATCH-',I1,' . ')
     106    FORMAT(16H@ASG,TZ SCRATCH-,I1,7H,16N . )
     108    FORMAT(8H@FREE,A ,I2,3H . )
     109    FORMAT(8H@ASG,TZ ,I2,'.,F/O/POS/64    . ')
     110    FORMAT('1INVALID UNIT FOR SETUP SUBROUTINE. RUN HALTS.')
     111    FORMAT(1X,7A6)
     112    FORMAT('1DYNAMICALLY ISSUED CONTROL STATEMENTS FOR ',A6,//)
     115    FORMAT('@USE ',I2,',FILE',I1,' . ')
            END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C*****************************************************************
C*                                                               *
C*      GROUP MEMBERSHIP PROGRAM                                 *
C*                                                               *
C*               PURPOSE  TO PRINT MEMBERSHIP AT VARIOUS         *
C*                        HIERARCHIAL GROUPING STAGES            *
C*                                                               *
C*****************************************************************
C
      DIMENSION LABEL(120),KO(7000),KOS(7000),        AVE(7000),KA(50)
       DIMENSION  BSTE(7000)
      DIMENSION IA(7000)
       DIMENSION IOPU(2)
         DIMENSION ITC(8)
      INTEGER BLANK
      DATA KNM,BLANK/'GRMBRS',' '/
       EQUIVALENCE (KO,KOS)
      CALL SETUP(10,15,26,0,0,0,0,0,0,0,0,JSEQ)
      KIN = 10
      IT = 15
      IR = 26
      JOUT = 6
       IOPU(1) = 6
       IOPU(2) = IR
       MBIT = 262144
C
C      READ CONTROL CARD
C
8     READ(5,100) INM,LAB1,LAB2,NLINE,IDSP,ITC,IPTP,KOPT
100   FORMAT(A6,X,A6,A2,X,I1,X,A6,X,8A6,X,I1,X,I1)
C
       IOS = 0
C
C
C
C      READ HEADINGS
C
  9    NLABS = NLINE*14
             IF ( NLABS.EQ.0) GO TO 10
             READ 102,(LABEL(K),K= 1,NLABS)
 102   FORMAT  (13A6,A2)
C
C      MAIN PROGRAM LOOP
C
   10  CONTINUE
C
      LOW = 1
C
      KLINE = 55
      IPTP = IPTP + 2
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
         IF(IPTP.GE.3) IPTP = 1
     14     REWIND KIN
                 READ (KIN) KK,(KA(K),K=1,KK   )
         IF(KA(1).EQ.LAB1.AND.KA(2).EQ.LAB2) GO TO 97
         WRITE(JOUT,106) KA(1),KA(2),BLANK,LAB1,LAB2
    106          FORMAT (33H TAPE AND CNTRL CRD FAIL TO MATCH ,
      1                / 8H TAPE = A6,A3,A6
      3                / 8H CARD = A6,A3,A6   )
         STOP
     97          KK = KA(4) + 1
             DO  98 L =1,KK
                 READ(KIN)  DUMMY
     98      CONTINUE
         REWIND IT
         KMI = KA(3) - 1
           NCASE = KA(3)
         WRITE(IT) KA(1), KA(2), NCASE
                 KK = KA(3)
          BIGBST=0.
             DO  99 L =1,KK
                 READ (KIN) DUMMY,KO1,DUMMY,KOS1,BSTE(L),AVE(L)
          IF(BSTE(L).GT.BIGBST) BIGBST=BSTE(L)
                 KOS(L) = KO1   +KOS1* MBIT
     99      CONTINUE
          IF(KOPT.EQ.0) GO TO 15
         DO 425 IX = 1, KMI
          BSTE(IX)=BSTE(IX)/BIGBST*100.
    425   AVE(IX)=AVE(IX)/BIGBST*100.
     15          LINE = KLINE+1
                 LPG  = 0
                 JSWT = 1
C   THIS USED TO BE SUBROUTINE 'ARAY'
         DO 208 KU = 1, NCASE
    201   K = 0
    202   IGRUP = NCASE
          NU = 1
          KIU = K
    203   K = K + 1
          KOS1 = KOS(K)/MBIT - 1
          IF(K.GT.NCASE) GO TO 206
          IF(KOS1.GT.KU) GO TO 204
          IF(KOS1.LT.KU) GO TO 206
    204   IF(KOS1.GE.IGRUP) GO TO 205
          IGRUP = KOS1
    205   NU = NU + 1
          GO TO 203
    206   IF(NU.LE.1) GO TO 207
          KO1 = MOD(KOS(KIU),MBIT)
          IF(IGRUP.GT.KU) GO TO 207
          IA(IGRUP) = KO1
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
207   IF(K.LE.NCASE) GO TO 202
208   CONTINUE
C  END OF 'ARAY'
           DO    500 KU = LOW,KHI
               GO TO (380,200,401),JSWT
  200              NGRUPS = 0
                   JSWT = 3
                   K      = 0
  210              IGRUP  = NCASE
                   NU     = 1
                   KIU    = K
  220              K      = K +1
                   KOS1   = KOS(K) / MBIT -1
               IF ( K.GT.NCASE) GO TO 270
               IF (KOS1.GT.KU) GO TO 240
               IF(KOS1.LT.KU) GO TO 270
                   NI   = NU
                   KJU = K
 240           IF (KOS1.GE.IGRUP) GO TO 260
                   IGRUP = KOS1
  260              NU     = NU + 1
               GO TO  220
  270          IF ( NU.LE.1) GO TO 400
                   KO1    = MOD(KOS(KIU),MBIT)
                   KO2    = MOD(KOS(KJU),MBIT)
                   NGRUPS = NGRUPS +1
                   LKJU   = KIU + NU - 1
                   LKIU = KJU - 1
               IF ( IGRUP.GT.KU) GO TO 350
                   NJ    = NU - NI
      IWW=IGRUP+1
      DO 7000 IW=IWW,NCASE
      IOW=IW
      IF(KO1.EQ.IA(IW))GO TO 2001
7000  CONTINUE
      KOO1=IGRUP
      GO TO 2002
2001  KOO1=IOW
2002  CONTINUE
       DO 600  JIM = 1,IPTP
       KOUT = IOPU(JIM)
      DO 3000 IW=IWW,NCASE
      IOW=IW
      IF(KO2.EQ.IA(IW))GO TO 3001
3000  CONTINUE
      KOO2=IGRUP
      GO TO 3002
3001  KOO2=IOW
3002  CONTINUE
                  WRITE (KOUT,321)
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
         1IGRUP,NU,BSTE(IGRUP),AVE(IGRUP),NI,KO1,KOO1,KIU,LKIU,
        .2NJ,KO2,KOO2,KJU,LKJU
  321    FORMAT(2H  ,I5,I6,2F9.2,I7,I6,I7,2I5,
        1/2H  ,29X,I7,I6,I7,2I5)
C   CREATE OUTPUT TAPE FOR DIAGRM
          IF(JIM.EQ.2) GO TO 619
        WRITE(IT) IGRUP,NU,BSTE(IGRUP),AVE(IGRUP),NI,KO1,KOO1,KIU,LKIU,NJ,
       1KO2,KOO2,KJU,LKJU
  619  CONTINUE
  600   CONTINUE
   360          LINE = LINE + 2
                GO TO 350
   400          CONTINUE
   401       IF ( K.LE.NCASE) GO TO 210
                JSWT = 1
  350      CONTINUE
        GO TO (500,380,380) ,JSWT
   380  IF (LINE.LE.KLINE) GO TO 395
                LPG = LPG +1
        LINE = 2
        DO 900  JIM = 1,IPTP
        KOUT =  IOPU(JIM)
       WRITE(KOUT,326) ITC, IDSP, LPG
  326  FORMAT(1H1,8A6,29X,A6,' PAGE ',I3,/)
        IF (LPG.GT.1) GO TO 900
             IF ( NLABS.EQ.0) GO TO 390
                WRITE (KOUT, 325) (LABEL(L), L = 1,NLABS)
   325          FORMAT ( 2X,13A6,A2)
   390          WRITE (KOUT, 324 ) KA(1),KA(2),KA(7),KA(8),KA(6),KA(3)
   324 FORMAT ( /            41H                          .....MEMBERSHIP AT EACH
      1                45HSTAGE OF HIERARCHIAL GROUPING ON.. STUDY.... ,A6,
      2 A2,/75X,11H FORMULA.. I1,
      3 /31X,9HNO. MBRS.,35X,11H CRITERION ,A3,
      4                    ,5HIMIZE,
      5           /    19X,16HAVERAGE        IN, 40X       11H MATRIX... ,A6,
      6 /10X,3HNO.,6X,8HOVERLAPS,4X,7HMERGING,9X,5HSTAGE,3X,
      7 5HKPATH,16X,10HCASES.... ,I4
      8 /6H STAGE,2X,7HMEMBERS,1X,7HBETWEEN,2X,6HWITHIN,1X,6HGROUPS,1X,
      9 5HGROUP,2X,6HFORMED,1X,4HFROM,2X,2HTO/)
        IF (JIM.EQ.1) LINE = LINE + NLINE + 8
  900   CONTINUE
   395        GO TO (200,500,401),JSWT
   500     CONTINUE
         REWIND KIN
          IF (IPTP.NE.2) GO TO 5
        CALL REPEND(LAB1,IDSP,ITC,JSEQ)
C
5       REWIND IT
        STOP
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      DOUBLE PRECISION IOPERB,IOPERC,IOPERD
      DIMENSION LEFT(7,53),HAND(7,1)
      DIMENSION  LROW(2004,5), LROWA(2004,5), IWEGHT(2000), IODSK(1003),
     1NSTGMS(2000), NSTGWS(2000), IBDYVL(2), KOMM(50), IMAGE(4)
      DIMENSION MSG(5)
      EQUIVALENCE  (LROW(1,1),LROWA(2,1)), (IWEGHT(1),LROW(1,5))
      EQUIVALENCE (LEFT,HAND), (LEFT,LROW)
      DATA  MIN,MAX,IBLANK,NZER,NTWO/'MIN','MAX',' ','0','2'/
      DATA  IBDYVL(1),IBDYVL(2),IROW,JROW/-1,200000,1,2/, NCKPT/0/
      DATA IMAGE/'@FREE 29 .  ',2*' '/
      DATA IWEGHT/2000*1/
      DATA MSG/' CODAP   GROUP   PREPROCESSING'/
      CALL SETUPG(KSID)
      CALL ERTRAN(6,IMAGE)
      ENCODE(8100,IMAGE) KSID
      LENH = 5
      CALL ERII('L',LENH,MSG,ICKOVR)
C  INITIALIZE SOME PROCESSING FLAGS.
      ASSIGN 4320 TO KROUTE
      ASSIGN 3440 TO INSERT
      ASSIGN 5520 TO JUMP
C  ASSIGN UNITS.
      KTIN = 10
      KTOT = 22
      KTOF = 15
      IO = 29
C  ALIGN  UNITS.
      REWIND KTIN
      REWIND KTOT
C  READ CONTROL CARD.
      READ(5,8010) ID1,ID2,LOP,MAXMIN,KOLOP
C  READ OVRLAP TAPE COMM.
      READ(KTIN) NUM,(KOMM(INDEXA),INDEXA = 1,NUM)
C  CHECK STUDY I.D. - ABORT FOR NO MATCH.
      IF(KOMM(1).EQ.ID1 .AND. KOMM(2).EQ.ID2) GO TO 100
      WRITE(6,8020) KOMM(1), KOMM(2),ID1,ID2
      CALL ERIIX
      CALL ERTRAN(2)
100   CONTINUE
C  EXTRACT DATA FROM COMM.
      NCASE = KOMM(3)
      MID = KOMM(6)
C  CALCULATE USEFUL QUANTITIES.
      NCASEH = NCASE/2 +1
      JAUG1 = NCASE/2 +2
      JAUG2 = NCASE/2 +3
      NCASP1 = NCASE + 1
      NCASP2 = NCASE + 2
      NCASP3 = NCASE + 3
      JOAL = NCASE - 1
```

LINE #   TEXT...

```
      C
             JERK = 1008
             DEFINE FILE 10(NCASE,JERK,U,NEXT)
      C
      C   SET SOME MAJOR BRANCH FLAGS.
      C
             IF(LOP.EQ.IBLANK .OR. LOP.EQ.NZER) GO TO 200
             ASSIGN  7520 TO LHEAD
             ASSIGN 7540 TO LINOP
             ASSIGN 7591 TO LINOUT
             LINSUB = 4
             LINES = 4
             GO TO 300
        200  CONTINUE
             ASSIGN  7530 TO LHEAD
             ASSIGN 7590 TO LINOP
             ASSIGN 7595 TO LINOUT
        300  CONTINUE
             IF(MAXMIN.EQ.IBLANK .OR. MAXMIN.EQ.NZER) GO TO 400
             ASSIGN  520 TO MSUBOP
             ASSIGN 1320 TO KROOP
             ASSIGN 3100 TO MMSWCH
             NSUB = 2
             MMWD = MIN
             GO TO 500
        400  CONTINUE
             ASSIGN  540 TO MSUBOP
             ASSIGN 1340 TO KROOP
             ASSIGN 3200 TO MMSWCH
             NSUB = 1
             MMWD = MAX
        500  CONTINUE
             IF(KOLOP.EQ.IBLANK .OR. KOLOP.EQ.NTWO) GO TO 600
             GO TO MSUBOP,(520,540)
        520  CONTINUE
             ASSIGN 4800 TO KOLAPS
             GO TO 560
        540  CONTINUE
             ASSIGN 4700 TO KOLAPS
        560  CONTINUE
             NGROUP = 1
             GO TO 700
        600  CONTINUE
             ASSIGN 4900 TO KOLAPS
             NGROUP = 2
        700  CONTINUE
      C
      C   WRITE RECORD 1 OF GROUP OUTPUT TAPE.
             WRITE(KTOT) ID1,ID2,MID,NGROUP,MMWD
      C
```

LINE #   TEXT...

```
C
C  ROW READ-IN ROUTINE
C
      INDECX = NCASE
      KSUB = 1
      ASSIGN 1200 TO IFLAGA
      ASSIGN 1300 TO IFLAGB
800   CONTINUE
C  READ AN OVERLAP ROW.
      READ(KTIN) (IODSK(INDEXA),INDEXA = 1,NCASEM)
C  ASSIGN INITIAL VALUES TO UPDATE POINTER AND AVERAGE-WITHIN.
      IODSK(JAUG1) = 1
      IODSK(JAUG2) = 100000
C  OBTAIN MATRIX ROW NUMBER.
      MTXROW = FLD(0,18,IODSK(1))
      GO TO 1000
C
C  ROW PACKER
900   CONTINUE
      IFLIPA = 2
      DO 960 INDEXB = 1,NCASP1
      INDEXC = (INDEXB + 1)/2
      IFLIPA = 3 -IFLIPA
      GO TO (920,940),IFLIPA
920   CONTINUE
      FLD(0,18,IODSK(INDEXC)) = LROWA(INDEXB,KSUB)
      GO TO 960
940   CONTINUE
      FLD(18,18,IODSK(INDEXC)) = LROWA(INDEXB,KSUB)
960   CONTINUE
      IODSK(JAUG1) = LROWA(NCASP2,KSUB)
      IODSK(JAUG2) = LROWA(NCASP3,KSUB)
C
C  DISK WRITER
1000  CONTINUE
      FIND(IO'MTXROW)
      WRITE(IO'MTXROW,ERR=7600) (IODSK(I),I = 1,JAUG2)
      GO TO IFLAGA,(1200,3460,6200)
C
C  DISK READER
1100  CONTINUE
      READ(IO'MTXROW,ERR=7600) (IODSK(I),I = 1,JAUG2)
C
C  ROW UNPACKER
1200  CONTINUE
      IFLIPA = 2
      DO 1260 INDEXB = 1,NCASP1
      INDEXC = (INDEXB + 1)/2
      IFLIPA = 3 - IFLIPA
      GO TO (1220,1240),IFLIPA
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
1220  CONTINUE
      LROWA(INDEXB,KSUB) = FLD(0,18,IODSK(INDEXC))
      GO TO 1260
1240  CONTINUE
      LROWA(INDEXB,KSUB) = FLD(18,18,IODSK(INDEXC))
1260  CONTINUE
      LROWA(NCASP2,KSUB) = IODSK(JAUG1)
      LROWA(NCASP3,KSUB) = IODSK(JAUG2)
      GO TO IFLAGB,(1300,3700,5700)
C
1300  CONTINUE
C  SET-UP FOR EXTRACTING ROW EXTREMUM
      ASSIGN 3000 TO IAWAY
1310  CONTINUE
      ASSIGN 1500 TO LDIAGL
      ASSIGN 1800 TO KMPARE
      GO TO KROOP,(1320,1340)
1320  CONTINUE
      ASSIGN 1700 TO KMPARE
1340  CONTINUE
      ASSIGN 2000 TO IXFSUB
      ASSIGN 2300 TO IPRSUB
      MSUB = 5
C
C
C  EXCHANGE BUBBLE EXTREMUM/COUNTEREXTREMUM EXTRACTOR
C
1400  CONTINUE
C  PLACE BOUNDARY VALUE INTO EXCHANGE BUBBLE.
      IXCHBB = IBDYVL(NSUB)
C
      DO 2200 INDEXD = NCASE,1,-1
C  IF A MATRIX ROW IS BEING SCANNED, IGNORE (I,I) ELEMENT.
      GO TO LDIAGL,(1500,1600)
1500  CONTINUE
      IF(INDEXD.EQ.LROWA(1,KSUB)) GO TO 2200
1600  CONTINUE
C  CHECK DELETE FLAG.
      IF(LROW(INDEXD,MSUB).LT.0) GO TO 2200
C  BY OPTION, COMPARE ELEMENT VALUE AND EXCHANGE VALUE.
      GO TO KMPARE,(1700,1800)
1700  CONTINUE
      IF(LROW(INDEXD,KSUB).LT.IXCHBB) GO TO 1900
      GO TO 2200
1800  CONTINUE
C  IF ELEMENT VALUE GREATER THAN EXCHANGE VALUE, GO TO NEXT STATEMENT.
      IF(LROW(INDEXD,KSUB).LE.IXCHBB) GO TO 2200
1900  CONTINUE
C  EXCHANGE THE VALUES.
      IXCHBB = LROW(INDEXD,KSUB)
```

LINE #   TEXT...

```
C  BY OPTION, ASSIGN SUBSCRIPTS.
      GO TO IXFSUB,(2000,2100)
2000  CONTINUE
      ISUBI = INDEXD
      GO TO 2200
2100  CONTINUE
      ISUBI = LROW(INDEXD,4)
2200  CONTINUE
C
C  BY OPTION, STORE  SUBSCRIPTS.
      GO TO IPRSUB,(2300,2400)
2300  CONTINUE
      ISUBJ = LROWA(I,KSUB)
      GO TO 2500
2400  CONTINUE
      ISUBJ = FLD(18,18,ISUBI)
      ISUBI = FLD(0,18,ISUBI)
2500  CONTINUE
      GO TO IAWAY,(3000,3300,5700,3350,5900)
3000  CONTINUE
C  DECREMENT ROW INPUT INDEX.
      INDECX = INDECX - 1
      INDEXX = NCASE - INDECX
C  STORE BEST VALUE AND SUBSCRIPTS FOR EACH ROW.
      LROW(INDEXX,3) = IXCHBB
      FLD(0,18,LROW(INDEXX,4)) = ISUBI
      FLD(18,18,LROW(INDEXX,4)) = ISUBJ
      IF(INDECX.GT.0) GO TO 800
C
C  NCASE ROWS SHOULD BE ON DISK, NCASE ENTRIES IN BEST TABLE.
C  CLOSE INPUT FILE.
      REWIND KTIN
      CALL FREE(KTIN)
C  WRITE CHECKPOINT HEADER.
      WRITE(6,8070)
C  SCAN BEST TABLE FOR EXTREMAL VALUE.
C  SET-UP
      ASSIGN 3300 TO IAWAY
3050  CONTINUE
      ASSIGN 1600 TO LDIAGL
      ASSIGN 1800 TO KMPARE
      GO TO MMSWCH,(3100,3200)
3100  CONTINUE
      ASSIGN 1700 TO KMPARE
3200  CONTINUE
      ASSIGN 2100 TO IXFSUB
      ASSIGN 2400 TO IPRSUB
      KSUB = 3
      MSUB = 3
      GO TO 1400
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 3300  CONTINUE
C  IXCHBB CONTAINS BEST VALUE, AND ISUBI AND ISUBJ CONTAIN BEST SUB-
C     SCRIPTS.
C  BEGIN PROCESSING BY STAGES.
       INDECX = NCASE - 1
       IPOINT = 1
 3350  CONTINUE
C  ADJUST IROW AND JROW SUBSCRIPTS.
       IF(ISUBI.LT.ISUBJ) GO TO 3400
       ITEMP = ISUBI
       ISUBI = ISUBJ
       ISUBJ = ITEMP
 3400  CONTINUE
C  ISUBI IS IROW ROW NUMBER, ISUBJ IS JROW ROW NUMBER.
C  RETRIEVE NEEDED ROW(S) FROM DISK.
       IF(LROWA(1,IROW).EQ.ISUBI) GO TO 3800
       IF(LROWA(1,IROW).EQ.ISUBJ) GO TO 3500
       GO TO INSERT,(3420,3440)
 3420  CONTINUE
C  INSERT OLD IROW BACK ONTO DISK.
       KSUB = IROW
       MTXROW = LROWA(1,IROW)
       ASSIGN 3460 TO IFLAGA
       GO TO 900
 3440  CONTINUE
       ASSIGN 3420 TO INSERT
 3460  CONTINUE
       ASSIGN 3800 TO ISLECT
       GO TO 3600
 3500  CONTINUE
C  CORRECT IROW AND JROW SUBSCRIPTS FOR LROW.
       IROW = JROW
       JROW = 3 - IROW
C  ONLY IROW IS NEEDED FROM DISK.
       ASSIGN 3900 TO ISLECT
 3600  CONTINUE
C  FETCH IROW FROM DISK.
       MTXROW = ISUBI
       KSUB = IROW
       ASSIGN 3700 TO IFLAGB
       GO TO 1100
 3700  CONTINUE
       GO TO ISLECT,(3800,3900)
 3800  CONTINUE
C  FETCH JROW FROM DISK.
       MTXROW = ISUBJ
       KSUB = JROW
       ASSIGN 3900 TO IFLAGB
       GO TO 1100
 3900  CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C  UPDATE ROWS AS REQUIRED.
       KSUBI = 2
       INDEXP = 2
 4000  CONTINUE
       INDEXU = LROWA(NCASP2,KSUBI)
       IF(INDEXU.EQ.IPOINT) GO TO 4300
C  ROW UPDATE NECESSARY.
C  SET-UP
       ASSIGN 4200 TO JMPBAK
       KSUBJ = KSUBI
 4100  CONTINUE
       ICOL = FLD(0,18,NSTGMS(INDEXU))
       JCOL = FLD(18,18,NSTGMS(INDEXU))
       IW = FLD(0,18,NSTGWS(INDEXU))
       JW = FLD(18,18,NSTGWS(INDEXU))
       IJW = IW + JW
       IJWH = IJW / 2
       GO TO 4600
 4200  CONTINUE
C  CONTINUE UPDATE UNTIL TABLE EXHAUSTED.
       INDEXU = INDEXU + 1
       IF(INDEXU.LT.IPOINT) GO TO 4100
 4300  CONTINUE
       KSUBI = 3 - KSUBI
       INDEXP = INDEXP - 1
C  IF THERE IS A SECOND ROW, CHECK IT ALSO.
       IF(INDEXP.GT.0) GO TO 4000
       GO TO KROUTE,(4320,5800)
 4320  CONTINUE
C
C  ROW UPDATE(S) COMPLETE - COLLAPSE ROWS.
C  SET-UP
       KSUBI = IROW
       KSUBJ = JROW
       IW = IWEGHT(ISUBI)
       JW = IWEGHT(ISUBJ)
       IJW = IW + JW
       IJWH = IJW / 2
       ASSIGN 4500 TO JMPBAK
       INDEXL = 1
 4400  CONTINUE
C  CHECK DELETE FLAG.
       IF(IWEGHT(INDEXL).LT.0) GO TO 4500
       ICOL = INDEXL
       JCOL = ICOL
       GO TO 4600
 4500  CONTINUE
C  CONTINUE COLLAPSE OVER ALL ACTIVE ELEMENTS.
       INDEXL = INDEXL + 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      IF(INDEXL.LE.NCASE) GO TO 4400
C  ROWS COLLAPSED - FLAG JROW FOR DELETION.
      IWEGHT(ISUBJ) = -1
C  GO TO CALCULATIONS AND OUTPUT SECTION.
      GO TO 5100
C
C  THE NESTING PLACE OF THE GREAT ELEMENT COLLAPSING ROUTINE
4600  CONTINUE
C  BRANCH TO COLLAPSE OPTION.
      GO TO KOLAPS,(4700,4800,4900)
4700  CONTINUE
      LROW(ICOL,KSUBI) = MAX0(LROW(ICOL,KSUBI),LROW(JCOL,KSUBJ))
      GO TO 5000
4800  CONTINUE
      LROW(ICOL,KSUBI) = MIN0(LROW( COL,KSUBI),LROW(JCOL,KSUBJ))
      GO TO 5000
4900  CONTINUE
      LROW(ICOL,KSUBI) = ( LROW(ICOL,KSUBI)*IW + LROW(JCOL,KSUBJ)*JW +
     1 IJWH ) / IJW
5000  CONTINUE
      GO TO JMPBAK,(4200,4500)
5100  CONTINUE
C  CALCULATE NEW AVERAGE-WITHIN FOR IROW.
C  NOTE USE OF DOUBLE PRECISION FOR INTEGER OVERFLOW.
      IOPERB = LROWA(NCASP3,IROW)*IW**2/IJW
      IOPERC = LROWA(NCASP3,JROW)*JW**2/IJW
      IOPERD = IXCHBB*IW*JW*2/IJW
      LROWA(NCASP3,IROW) = (IOPERB + IOPERC + IOPERD + IJWH)/IJW
C  CONVERT AVERAGE-WITHIN AND BEST CRITERION VALUE FROM ROUNDED, BIASED
C     INTEGER TO REAL.
      AVGN = LROWA(NCASP3,IROW)
      BEST = IXCHBB
      AVGN = AVGN/1000.
      BEST = BEST/1000.
C  WRITE A RECORD TO TEMPORARY OUTPUT FILE.
      WRITE(KTOT) INDECX,ISUBI,ISUBJ,IW,JW,BEST,AVGN
C  MAKE ENTRY INTO ROW UPDATE TABLES.
      FLD(0,18,NSTGRS(IPOINT)) = ISUBI
      FLD(18,18,NSTGRS(IPOINT)) = ISUBJ
      FLD(0,18,NSTGWS(IPOINT)) = IW
      FLD(18,18,NSTGWS(IPOINT)) = JW
C  POLL FOR CHECKPOINT.
      MODD = MOD(IPOINT,10)
      IF(MODD.NE.0 .AND. ICKOVR.EQ.0) GO TO 5560
      CALL ERTRAN(9,KDATE,KTIME)
      DECODE(8080,KTIME) LHH,LMM,LSS
      DECODE(8090,KTIME) LH,LM,LS
      LATE = LH*3600 + LM*60 + LS
      GO TO JUMP,(5520,5540)
5520  CONTINUE
```

LINE # TEXT...

```
           IRLY = LATE
           ASSIGN 5540 TO JUMP
     5540  CONTINUE
           LAPSE = LATE - IRLY
           LAPSE = IABS(LAPSE)
           IF(ICKOVR.NE.0) GO TO 5550
           IF(LAPSE.LT.3000) GO TO 5560
     5550  ICKOVR = 0
           CALL ERTHAN(6,IMAGE)
           CALL ERTHAN(9,KDATE,KTIME)
           DECODE(8080,KTIME) LHL,LML,LSL
           DECODE(8090,KTIME) LH,LM,LS
           LATE = LH*3600 + LM*60 + LS
           IRLY = LATE
           NCKPT = NCKPT + 1
           WRITE(6,8110) NCKPT,LHH,LMM,LSS,LHL,LML,LSL,INDECX
     5560  CONTINUE
     C
     C  START NEXT STAGE.
           INDECX = INDECX - 1
           CALL ERTHAN(9,KDATE,KTIME)
           FLD(0,12,LHH) = FLD(0,12,KTIME)
           FLD(0,12,LMM) = FLD(12,12,KTIME)
           FLD(0,12,LSS) = FLD(24,12,KTIME)
           IF(INDECX.EQ.0) GO TO 7500
           IPOINT = NCASE - INDECX
           ENCODE(8170,MSG) LHH,LMM,LSS,IPOINT,JOAL
     8170  FORMAT(A2,':',A2,':',A2,'    GROUP ',I4,'/',I4,4X)
     C  GENERATE WEIGHT AND UPDATE POINTER FOR NEW IROW.
           IWEGHT(ISUBI) = IW + JW
           LROWA(NCASP2,IROW) = IPOINT
     C  FLAG DELETED ENTRIES IN BEST TABLE, UPDATING THE REMAINDER.
     C
     C  SET FLAGS, POINTERS, AND INITIAL VALUES.
     C  NOTE THAT HERE ISUBI IS IROW ROW NUMBER, WHILE AFTER FIRST ROW
     C    EXTREMUM EXTRACTION, ISUBI IS ELEMENT COLUMN COORDINATE.
           KOMPI = ISUBI
           KOMPJ = ISUBJ
           KSUB = JROW
           ASSIGN 5700 TO IFLAGB
           ASSIGN 5800 TO KROUTE
           ASSIGN 5900 TO IAWAY
           ASSIGN 6000 TO IRFLAG
           ASSIGN 6200 TO IFLAGA
           INDEXR = 1
     5600  CONTINUE
           INDEXG = INDEXR
     C  CHECK DELETE FLAG.
           IF(LROW(INDEXR,3).LT.0) GO TO 6300
     C  CHECK FOR IROW OR JROW SUBSCRIPTS.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      IF(FLD(0,18,LROW(INDEXR,4)).NE.KOMPI .AND. FLD(0,18,LROW(INDEXR,4)
     1).NE.KOMPJ .AND. FLD(18,18,LROW(INDEXR,4)).NE.KOMPI .AND.
     2 FLD(18,18,LROW(INDEXR,4)).NE.KOMPJ) GO TO 6300
C  IF THIS IS JROW, FLAG FOR DELETION.
      IF(FLD(18,18,LROW(INDEXR,4)).NE.KOMPJ) GO TO 5650
      LROW(INDEXR,3) = -1
      GO TO 6300
5650  CONTINUE
C  IF THIS IS IROW, ALREADY UPDATED AND STILL IN CORE, BYPASS FETCH, UP-
C     DATE, AND RETURN OPERATIONS.
      IF(FLD(18,18,LROW(INDEXR,4)).EQ.KOMPI) GO TO 5850
5660  CONTINUE
C  FETCH A ROW FROM DISK.
      MTXROW = FLD(18,18,LROW(INDEXG,4))
      GO TO 1100
5700  CONTINUE
C  UPDATE ROW.
      KSUBI = JROW
      INDEXP = 1
      GO TO 4000
5800  CONTINUE
C  EXTRACT ROW EXTREMUM.
      GO TO 1310
5850  CONTINUE
      ASSIGN 6100 TO IRFLAG
      KSUB = IROW
      GO TO 1310
5900  CONTINUE
C  ENTER EXTREMUM INTO BEST TABLE ELEMENT.
      LROW(INDEXG,3) = IXCHBB
      FLD(0,18,LROW(INDEXG,4)) = ISUBI
      FLD(18,18,LROW(INDEXG,4)) = ISUBJ
      GO TO IRFLAG,(6000,6100)
6000  CONTINUE
C  ENTER UPDATE POINTER.
      LROWA(NCASP2,JROW) = IPOINT
C  RETURN ROW TO DISK.
      GO TO  900
6100  CONTINUE
C  RESTORE DISK FETCH, UPDATE, AND RETURN OPERATIONS.
      ASSIGN 6000 TO IRFLAG
      KSUB = JROW
6200  CONTINUE
C  IF ROW ISUBI IS NOT IROW, AND THE BEST TABLE ENTRY IS A NON-EQUAL
C     SYMMETRIC ELEMENT, THE ENTRY MUST BE UPDATED.
      IF(ISUBI.EQ.KOMPI .OR. FLD(0,18,LROW(ISUBI,4)).NE.ISUBJ .OR.
     1LROW(ISUBI,3).EQ.IXCHBB) GO TO 6300
      INDEXG = ISUBI
      GO TO 5660
6300  CONTINUE
```

LINE #   TEXT...

```
C  CONTINUE PROCESS UNTIL ALL ROWS EXAMINED.
      INDEXR = INDEXR + 1
      IF(INDEXR.LE.NCASE) GO TO 5600
C  RESET ROW UPDATE EXIT FLAG.
      ASSIGN 4320 TO KROUTE
C  SEARCH BEST TABLE FOR NEXT ROW(S) TO BRING FROM DISK AND PROCESS.
C  SET-UP
      ASSIGN 3350 TO IAWAY
      GO TO 3050
C
C  TRANSFER OUTPUT TO TAPE FILE.
7500  CONTINUE
      ENDFILE KTOT
      REWIND KTOT
      CALL SETGRP
      REWIND KTOF
      READ(KTOT) ID1,ID2,MID,NGROUP,MMWD
      WRITE(KTOF) ID1,ID2,MID,NGROUP,MMWD
C  BY OPTION, WRITE PRINTER HEADING.
      GO TO LHEAD,(7520,7530)
7520   CONTINUE
      WRITE(6,8030) MID,MMWD,NGROUP
      NPAGE = 2
      JUMPSD = 1
7530  CONTINUE
      NCASE = NCASE - 1
      DO 7590 INDEXI = 1,NCASE
      READ(KTOT) INDECX,ISUBI,ISUBJ,IW,JW,BEST,AVGN
      WRITE(KTOF) INDECX,ISUBI,ISUBJ,IW,JW,BEST,AVGN
C  BY OPTION, WRITE STAGE INFO ON PRINTER.
      GO TO LINOP,(7540,7590)
7540  CONTINUE
C  BRANCH FOR HALF PAGE PRINTOUT.
      GO TO(7545,7570),JUMPSD
7545  CONTINUE
C  CHECK PAGE (LEFT COLUMN) OVERFLOW.
      IF(LINES.GT.53) GO TO 7560
7550  CONTINUE
C  ACCUMULATE LEFT HAND PRINT COLUMN.
      LEFT(1,LINES) = INDECX
      LEFT(2,LINES) = ISUBI
      LEFT(3,LINES) = ISUBJ
      LEFT(4,LINES) = IW
      LEFT(5,LINES) = JW
      HAND(6,LINES) = BEST
      HAND(7,LINES) = AVGN
      LINES = LINES + 1
      GO TO 7590
7560  CONTINUE
C  SET RIGHT HAND COLUMN FLAG.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      JUMP50 = 2
 7570 CONTINUE
C  CHECK PAGE (RIGHT COLUMN) OVERFLOW.
      IF(LINSUB.GE.LINES) GO TO 7580
C  WRITE A LINE.
      WRITE(6,8040) (LEFT(INDEX1,LINSUB),INDEX1 = 1,5),(HAND(INDEX1,
     1 LINSUB),INDEX1 = 6,7),INDECX,ISUBI,ISUBJ,IW,JW,BEST,AVGN
      LINSUB = LINSUB + 1
      GO TO 7590
 7580 CONTINUE
C  PAGE ADVANCE AND HEADER
      WRITE(6,8050) NPAGE
      LINES = 1
      LINSUB = 1
      JUMP50 = 1
      NPAGE = NPAGE + 1
      GO TO 7550
 7590 CONTINUE
C  BRANCH TO CLOSE OUT LEFTHAND PRINT COLUMN.
      GO TO LINOUT,(7591,7595)
 7591 CONTINUE
      IF(LINSUB.GE.LINES) GO TO 7595
      LINES = LINES - 1
      DO 7593 INDEX1 = LINSUB,LINES
      WRITE(6,8040) (LEFT(INDEX1,INDEX1),INDEX1 = 1,5),
     1 (HAND(INDEX1,INDEX1),INDEX1 = 6,7)
 7593 CONTINUE
 7595 CONTINUE
C  PROGRAM CLOSEOUT
      ENDFILE KTOF
      REWIND KTOF
      CALL ENDGRP
      WRITE(6,8060) NCASE,ID1,ID2
      CALL ERIIX
      STOP
C  INTERNAL I/O ERROR
 7600 CONTINUE
      WRITE(6,8000) MTXROW
      CALL ERIIX
      CALL ERTRAN(2)
 8000 FORMAT(1H1,4X,'ATTEMPT TO RANDOMLY READ/WRITE RECORD BEYOND ALLOCA
     1TED DISK LIMITS.'/5X,'ROW NUMBER: ',O12,' .',5X,'GROUP ABORTS.')
 8010 FORMAT(7X,1A6,1A2,1X,1A1,1X,1A1,1X,1A1)
 8020 FORMAT(1H1,4X,'OVRLAP TAPE STUDY I.D. DOES NOT MATCH CONTROL CARD
     1STUDY I.D. .'/5X,'OVRLAP TAPE STUDY I.D. IS',1A6,1A2,'AND CONTROL
     2CARD STUDY I.D. IS',1A6,1A2,' .'/5X,'RUN ABORTS.')
 8030 FORMAT(1H1,2X,'OVERLAP MATRIX ID = ',1A6,29X,'GROUP PAGE    1'/3X,
     11A3,'IMIZE'/3X,'GROUP OPTION = ',I11/3X,'STAGE',2X,'IBEST',2X,'JBE
     2ST',2X,'MBRS I',3X,'MBRS J',4X,'BEST',5X,'AVG',20X,'STAGE',2X,'IBE
     3ST',2X,'JBEST',2X,'MBRS I',3X,'MBRS J',4X,'BEST',5X,'AVG'//)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
8040  FORMAT(2(4X,I4,3X,I14,3X,I14,3X,I14,5X,I14,2X,1F7.2,2X,1F7.2,T73))
8050  FORMAT(1H1,57X,'GROUP PAGE ',I13 / 2(3X,'STAGE',2X,'IBEST',2X,'JBE
     1ST',2X,'MBRS I',3X,'MBRS J',4X,'BEST',5X,'AVG',T73)/)
8060  FORMAT(1H1,2X,'GROUP HAS COMPLETED ',I14,' STAGES FOR STUDY ',1A6,
     11A2,' .'/3X,'GROUP ENDS.')
8070  FORMAT('+         CHECKPOINT TAKEN'/7X,'FROM',2X,'TIME',3X,'TO',5X,
     1 'STAGE')
8080  FORMAT(3A2)
8090  FORMAT(3I2)
8100  FORMAT('WCKPT,P    GP',A4,'CKPT, . ')
8110  FORMAT(1X,I3,2(2X,A2,1H:,A2,1H:,A2),4X,I4)
C
      SUBROUTINE SETUPG(ISTUDY)
C  THIS IS A SPECIAL SUBROUTINE FOR THE GROUP PROGRAM
C  IT IS DESIGNED TO ASSIGN OUTPUT TAPES ONLY AFTER ALL
C  CALCULATIONS ARE COMPLETED AND RECORDED ON MASS STORAGE.
      DIMENSION NAM1(3), NAM2(3), IMAGE(8), ID(2), ICON(3)
      DATA LD,LI/'D','I'/
      DATA ID/'ID      '/
      IUSE = 10
      ISC2 = 22
      IOUT = 15
      READ(5,100) IDPROG, NAM1, NAM2
      WRITE(6,101) IDPROG
      CALL ASGA(NAM1,IUSE)
      ENCODE(102,ISTUDY) NAM2(1)
      ASSIGN 7 TO IPICK
      ASSIGN 3 TO JPICK
      ENCODE(120,ICON) ISTUDY
1     ENCODE(104,IMAGE) ICON
      CALL ERTRAN(6,IMAGE)
      ENCODE(108,IMAGE) ICON
      CALL ERTRAN(6,IMAGE)
      CALL DKSTAT(ID,IC1,IC2,IFLAG)
      ENCODE(112,IMAGE) ICON
      CALL ERTRAN(6,IMAGE)
      IF:IFLAG.EQ.1) GO TO IPICK,(7,11)
      GO TO JPICK,(3,5)
3     IIF1 = LD
      ENCODE(122,IMAGE) ICON
      WRITE(6,106) (IMAGE(I),I=1,4)
      CALL GETFIL(IMAGE)
      GO TO 9
5     IIF2 = LD
      ENCODE(123,IMAGE) ICON
      WRITE(6,106) (IMAGE(I),I=1,4)
      CALL GETFIL(IMAGE)
      GO TO 13
7     ENCODE(103,IMAGE) ISTUDY
      CALL GETFIL(IMAGE)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      IIF1 = LD
      ENCODE(112,IMAGE) ICON
      CALL ERTRAN(6,IMAGE)
      ENCODE(122,IMAGE) ICON
      WRITE(6,106) (IMAGE(I),I=1,4)
      CALL GETFIL(IMAGE)
  9   ASSIGN 11 TO IPICK
      ASSIGN 5 TO JPICK
      ENCODE(124,ICON) ISTUDY,ISC2
      GO TO 1
 11   ENCODE(109,IMAGE) ISTUDY, ISC2
      CALL GETFIL(IMAGE)
      IIF2 = LD
      ENCODE(112,IMAGE) ICON
      CALL ERTRAN(6,IMAGE)
      ENCODE(122,IMAGE) ICON
      WRITE(6,106) (IMAGE(I),I = 1,4)
      CALL GETFIL(IMAGE)
 13   ENCODE(111,IMAGE) ISC2, ISTUDY, ISC2
      WRITE(6,106) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      WRITE(6,113)
      RETURN
      ENTRY SETGRP
      WRITE(6,107)
      IF(NAM2(1).NE.'SCHATC') GO TO 14
      ENCODE(121,IMAGE)
      WRITE(6,106) (IMAGE(I),I = 1,3)
      CALL ERTRAN(6,IMAGE)
      ENCODE(105,IMAGE)
      WRITE(6,106) (IMAGE(I),I = 1,4)
      CALL ERTRAN(6,IMAGE)
      ENCODE(114,IMAGE) IOUT
      WRITE(6,106) (IMAGE(I),I = 1,4)
      CALL ERTRAN(6,IMAGE)
      RETURN
 14   CALL ASGC(NAM2,IOUT)
      RETURN
      ENTRY ENDGRP
      WRITE(6,113)
      ENCODE(110,IMAGE) IIF2,ISTUDY,ISC2
      WRITE(6,106) (IMAGE(I),I=1,4)
      CALL ERTRAN(6,IMAGE)
      ENCODE(115,IMAGE) IIF1,ISTUDY
      WRITE(6,106) (IMAGE(I),I=1,4)
      CALL ERTRAN(6,IMAGE)
      RETURN
100   FORMAT(A6,1X,2(A6,A4,A1,1X))
101   FORMAT('1DYNAMICALLY ISSUED CONTROL STATEMENTS 7OR ',A6,//)
102   FORMAT(A4)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

```
LINE #   TEXT...

 103     FORMAT('@ASG,CPZ GP',A4,'CKPT.,F/0/POS/25 . ')
 104     FORMAT('@ASG,YZ',3X,3A6)
 105     FORMAT('@ASG,TZ SCRATCH-GP.,16N . ')
 106     FORMAT(1X,8A6)
 107     FORMAT('1 GROUP COPYING TO OUTPUT TAPE',//)
 108     FORMAT('@USE  ID.,',3A6)
 109     FORMAT('@ASG,UZ  ',A4,'-',I2,'.,F/0/POS/20 . ')
 110     FORMAT('@FREE,',A1,2X,A4,'-',I2,'. . ')
 111     FORMAT('@USE ',I2,',',A4,'-',I2,'. . ')
 112     FORMAT('@FREE',5X,3A6)
 113     FORMAT(1H1)
 114     FORMAT('@USE ',I2,',SCRATCH-GP. . ')
 115     FORMAT('@FREE,',A1,'   GP',A4,'CKPT. . ')
 120     FORMAT('GP',A4,'CKPT. . ')
 121     FORMAT('@FREE,A 15 . ')
 122     FORMAT('@ASG,AZ',3X,3A6)
 123     FORMAT('@ASG,AZ ',2X,3A6)
 124     FORMAT(A4,'-',I2,'. . ',7X)
         END
```

GRPDIF GROUP DIFFERENCES PROGRAM

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      DIMENSION PMP(2000,3),APT(2000,3),AMP(2000,6),SKP(28),
     1 IRY(14880),KTIT(465,43),OPU(2),GREPID(2),
     2        TC(6),IGM(2),ISEQ(2),IGL(8,2),GSID(2),PI(7),
     3 IJA(50),JA(50),IPNI(4),LTAB(956),FFT(14,6),GM(2),
     4        IPN(4),P2(6)
C
      EQUIVALENCE(IRY(1),PMP(1,1),KTIT(1,1),AMP(1,1),SKP(28)),
     1           (IRY (6001),APT(1,1))
     2           (IRY(12001),IJA(1)),(IPN,IPNI),
     3           (IRY(12101),IGL(1,1))
C
      INTEGER    OPU,RDR,SPID
C
      DATA    IPNI/24HEND OF BLOCK DATA END   /
      DATA    KRD/6HGRPDIF/
      DATA    ILM/0/
      DATA    MBIT/262144/
      DATA    MCNT/40/
      DATA    LM/0/
C
      DATA TC/*DIFFERENCES   BETWEEN          AND              */
C
      CALL SETUP(10,27,26,0,0,0,20,21,0,0,0,LSEQ)
      IPT = 10
      IPR = 6
      IO = 27
      IR = 26
      IT = 20
      ITMP = 21
      RDR = 5
      OPU(1) = 6
      OPU(2) = IR
      CALL RESET
      LMAX = 55
       REWIND IO
       REWIND  IPT
       REWIND  IT
C
C      READ COMMUNICATION REGION
C
       READ (IPT) K,(JA(I),I=1,K)
       NTASK = JA(12)
      IAA1 = JA(1)
      IAA2 = JA(2)
C
C      SPACE TAPE TO DUTY-TASK TITLES
C
  100  READ (IPT) K,IRY(1),IRY(2)
       IF (IRY(1).NE.IPNI(1).AND.IRY(2).NE.IPNI(3)) GO TO 100
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C       READ AND BUILD NEW TITLE LOCATOR TABLE
C
  150  DO 200 I= 1,MCNT
       READ (IPT) K,(KTIT(J,I),J=1,K)
       IF (KTIT(1,I).EQ.IPN1(1).AND.KTIT(2,I).EQ.IPN(3)) GO TO 250
       KL = I
  200  CONTINUE
       WRITE (IT) KL,((KTIT(J,L),J=1,465),L=1,KL)
       GO TO 150
  250  IF(KL.LE.2) GO TO 260
       I= KL -2
       WRITE (IT) I,((KTIT(J,L),J= 1,465),L=1,I)
  260  CONTINUE
       REWIND  IT
       DO 270 I=1,3
       BACKSPACE IPT
  270  CONTINUE
C
C
       READ(IPT) K,(LTAB(I),I=1,K)
       REWIND IPT
       DO 280 J =1,NTASK
       K = LTAB(J)/MBIT
       KK= MOD(LTAB(J),MBIT)
       IF (K.EQ.I) GO TO 280
       I = K/MCNT +1
       LTAB(J) = I*MBIT +KK+465*(K-1)
  280  CONTINUE
C
C       READ AND CHECK HEADING CARDS
C
  300  READ(RDR,901) JRD, ISID, ISID2, KL
       CALL PRINTC
       IF(ISID.EQ.IAA1.AND.ISID2.EQ.IAA2) GO TO 303
       CALL ERRMSG(6,'TAPE-ID DOES NOT MATCH CONTROL CARD')
       WRITE(IPR,905) IAA1, IAA2
  905  FORMAT(' TAPE-ID IS ',A6,A2)
  302  READ(RDR,902,END=1000) IEND
       CALL PRINTC
       GO TO 302
  303  IF(KL.LE.6) GO TO 304
       CALL ERRMSG(6,'NUMBER OF HEADING CARDS EXCEEDS 6.')
       GO TO 302
  304  IF(KL.EQ.0) GO TO 401
       DO 305 J = 1, KL
       READ(RDR,902) (FFT(I,J),I=1,14)
       CALL PRINTC
  305  CONTINUE
       GO TO 401
  400  CALL RESET
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 401    READ(RDR,903,END=1000) BVAL, KVAL, KSRT, IOS, SPID
        CALL PRINTC
        IF(KVAL.EQ.0) KVAL = 3
        IF(KSRT.EQ.0) KSRT = 1
        IF(KSRT.LE.5) GO TO 308
        CALL ERRMSG(5,'INVALID SORT OPTION. ASSUMED 1')
 308    CONTINUE
        IF(KSRT.EQ.3) BVAL = .01
        IF(BVAL.EQ.0.0) BVAL = .01
        IF(IOS.EQ.0) IOS = 2
        IF(KVAL.LE.3) GO TO 122
        KVAL = 3
        GO TO 122
 306    CALL ERRMSG(4,'THIS REPORT IS BYPASSED')
        READ(RDR,902) IEND
        CALL PRINTC
        READ(RDR,902) IEND
        CALL PRINTC
        GO TO 400
 122    IF(KVAL.EQ.3) GO TO 123
        IF(KVAL.EQ.2) GO TO 121
        IF(KSRT.EQ.2.OR.KSRT.EQ.5) GO TO 307
        GO TO 123
 121    IF(KSRT.EQ.1.OR.KSRT.EQ.4) GO TO 307
        GO TO 123
 307    CALL ERRMSG(7,'INVALID PRINT/SORT OPTION COMBINATION')
        GO TO 306
   123  GO TO (124,125,126),KVAL
   124   KX=1
         KXX=3
         KLL=10
         GO TO 127
   125   KX=4
         KXX=6
         KLL=10
         GO TO 127
   126   KX=1
         KXX=6
         KLL=13
  127   CONTINUE
        IF(IOS.LE.2) GO TO 309
        IOS = 1
 309    DO 310 I = 1, 2
        READ(RDR,904) GSID(I),GRPID(I),(IGL(J,I),J=1,8),ISEQ(I)
        CALL PRINTC
 310    CONTINUE
 C
 C       SEARCH FOR DATA FROM BINARY JOB DEC. TAPE.
  500    J= 0
          I= 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
           IF (ISEQ(1).GT.ISEQ(2)) I= 2
  510    READ(IO,END=520) XDSP,ITASK,(PMP(K,1),K=1,ITASK),IK,(PMP(K,I),
        1 K=1,IK),
        2JSEQ,L,(IJA(K),K=1,L),(APT(K,I),K=1,ITASK),(PMP(K,I),K=1,ITASK)
           IF (JSEQ.GT.ISEQ(I)) GO TO 520
           IF (JSEQ.NE.ISEQ(I)) GO TO 510
           GREPID(I) = XDSP
          IF(IJA(35).NE.'JOBGRP'.AND.IJA(35).NE.'JOBSPC'.AND.IJA(35).NE.
        1 'SAMSEL') GO TO 583
          ENCODE(888,IGL(1,I)) (IJA(LG),LG=37,44)
 888      FORMAT(8A6)
 583      CONTINUE
           IGM(I) = IJA(17)
           J = J+1
           I = MOD(I,2)+1
          GO TO (510,550), J
  520     WRITE (IPR,16) I
  16      FORMAT (37H0CANNOT FIND RECORD FOR REQUEST CARD ,I4,
        1          32HIN THE ABOVE SET,REPORT SKIPPED.,/)
          GO TO 400
  550     REWIND IO
 C
 C        CALCULATE DIFFERENCE BETWEEN DESCRIPTIONS
 C
          GM(1) = IGM(1)
          GM(2) = IGM(2)
          DO 620  I = 1,NTASK
            DO 610 K = 1,2
             APT(I,K) = APT(I,K)/GM(K)
             PMP(I,K) =(PMP(I,K)/GM(K))*100.0
  610       CONTINUE
            PMP(I,3) = PMP(I,1)- PMP(I,2)
            APT(I,3) = APT(I,1)- APT(I,2)
  620     CONTINUE
 C
 C        PLACE SORTED VALUES ON TAPE TO MAKE ROOM FOR TITLES.
 C
          REWIND ITMP
          VAL1 = -MBIT
          IF (KSRT.GT.3) VAL1 = MBIT
          LN = MOD(KSRT,3)
          IF (KSRT.EQ.3) LN=1
          LM = LN+ 2*LN
          DO 800 K= 1,NTASK
           VAL = VAL1
           DO 780 I =1,NTASK
           GO TO (720,720,730,710,710),KSRT
  710      IF (AMP(I,LM).GE.VAL) GO TO 780
  711      VAL = AMP(I,LM)
           GO TO 740
```

193

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 720    IF (AMP(I,LM).LE.VAL) GO TO 780
        GO TO 711
 730    IQ = K
        VAL = MBIT
        GO TO 790
 740    IQ = 1
 780    CONTINUE
 790   WRITE (ITMP) IQ,PMP(IQ,2),APT(IQ,2),PMP(IQ,1),APT(IQ,1)
     1                  PMP(IQ,3),APT(IQ,3),VAL
       AMP(IQ,LM)= VALI
 800   CONTINUE
       REWIND ITMP
C
C      PRINT PAGE HEADINGS
C
 810   CONTINUE
       LPAGE = 1
       DO 820  I=1,IOS
           IUT = OPU(I)
           WRITE (IUT,9000)  GREPID(1),GREPID(2),SPID,LPAGE
 9000      FORMAT (20H1DIFFERENCE BETWEEN ,A6,5H AND ,A6,41X,A6,
     1             6H PAGE , I3/)
      IF(KL.GT.0) WRITE (IUT,9001)  ((FFT(J,K),J=1,14),K=1,KL)
 9001      FORMAT ( 1H ,13A6,A2 )
           WRITE (IUT,9002 ) ((GREPID(K),IGM(K),(IGL(J,K),J=1,8)),K=1,2)
 9002      FORMAT( 1X,A6,11H MEMBERS = ,I4,1X,8A6)
       GO TO (812,813,814)KVAL
  812  WRITE(IUT,9030) GREPID(1),GREPID(2),GREPID(2),GREPID(1)
 9030  FORMAT(1H0,19X,41HPERCENT MEMBERS PERFORMING---DIFFERENCE, ,A6,
     1 7H MINUS ,A6,23(1H.),/,20X,A6,27H PERCENT MEMBERS PERFORMING,
     2 42(1H.),7X,1H.,/,20X,A6,27H PERCENT MEMBERS PERFORMING,34(1H.),
     3 2(7X,1H.),/,79X,3(7X,1H.),/,11X,5HD-TSK,23X,10HTASK TITLE,30X,
     4 3(7X,1H.))
       GO TO 820
  813  WRITE(IUT,9031) GREPID(1),GREPID(2),GREPID(2),GREPID(1)
 9031  FORMAT(1H0,19X,56HAVERAGE PERCENT TIME BY ALL GROUP MEMBERS---DIF
     1FERENCE, ,A6,7H MINUS ,A6,8(1H.),/,20X,A6,42H AVERAGE PERCENT TIME
     2 BY ALL GROUP MEMBERS,27(1H.),7X,1H.,/,20X,A6,42H AVERAGE PERCENT
     3TIME BY ALL GROUP MEMBERS,19(1H.),2(7X,1H.),/,79X,3(7X,1H.),/,11X,
     4 5HD-TSK,23X,10HTASK TITLE,30X,3(7X,1H.))
       GO TO 820
  814  WRITE(IUT,9032)GREPID(1),GREPID(2),GREPID(2),GREPID(1),
     1                 GREPID(1),GREPID(2),GREPID(2),GREPID(1)
 9032  FORMAT(1H0,19X,56HAVERAGE PERCENT TIME BY ALL GROUP MEMBERS---DIF
     1FERENCE, ,A6,7H MINUS ,A6,35 (1H.),/,20X,A6,42H AVERAGE PERCENT TI
     2ME BY ALL GROUP MEMBERS,54(1H.),7X1(1H.),/20XA6,42H AVERAGE PERCEN
     3T TIME BY ALL GROUP MEMBERS,46(1H.),2(7X,1H.),/,20X,41HPERCENT MEM
     4BERS PERFORMING---DIFFERENCE, ,A6,7H MINUS ,A6,23(1H.),3X,3(7X,1H.
     5),/,20X,A6,27H PERCENT MEMBERS PERFORMING,42(1H.),7X,1H.,3X,
     6 3(7X,1H.),/,20X,A6,27H PERCENT MEMBERS PERFORMING,34(1H.),
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      7 2(7X,1H.),3X,3(7X,1H.),/,79X,3(7X,1H.),3X,3(7X,1H.),/,11X,
      8 5HD-TSK,23X,10HTASK TITLE,30X,3(7X,1H.),3X,3(7X,1H.))
 820  CONTINUE
C
C     READ DATA BACK FROM TAPE AND PRINT
C
      LCT=KLL+KL
      ILM = 0
      IFIRST=0
      DO 900  KJ = 1,NTASK
      READ (ITMP) IQ,(P1(I),I=1,7)
        IF (ABS(P1(7)).LT.0.01) GO TO 900
      IF(ABS(P1(7)).GE.BVAL) GO TO 856
      IF(IFIRST.NE.0) GO TO 900
      IFIRST=1
      LCT=LCT+6
      IF(LCT.LT.LMAX) GO TO 827
     LCT = 2
      LPAGE=LPAGE+1
      DO 825 J=1,IOS
      IUT=OPU(J)
      WRITE(IUT,9000) GREPID(1),GREPID(2),SPID,LPAGE
 825  CONTINUE
 827  DO 830 J=1,IOS
      IUT=OPU(J)
      WRITE(IUT,9049)
9049  FORMAT(1H0,104(1H*),/,  29X,46HTASKS OMITTED WHERE THE ABSOLUTE D
     1IFFERENCE IN)
      IF(KSRT.EQ.1.OR.KSRT.EQ.4) GO TO 828
      WRITE(IUT,9050) BVAL
9050  FORMAT(29X,39HAVERAGE PERCENT TIME SPENT IS LESS THAN,F7.2,/,1H ,
     1 104(1H*),/)
      GO TO 830
 828  WRITE(IUT,9051) BVAL
9051  FORMAT(29X,39HPERCENT MEMBERS PERFORMING IS LESS THAN,F7.2,/,1H ,
     1 104(1H*),/,)
 830  CONTINUE
     LCT = LCT + 3
      GO TO 900
 856    LM =LTAB(IQ)/MBIT
        LN = MOD(LTAB(IQ),MBIT)
        IF (ILM-LM) 857 ,870,858
 857    IM = LM-ILM
        GO TO 860
 858    IM = ILM-LM+1
        DO 859 KM = 1,IM
         BACKSPACE  IT
 859    CONTINUE
        IM = 1
 860    DO 861 KM = 1,IM
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          READ (IT) I,((KTIT(J,L),J=1,465),L=1,I)
 861   CONTINUE
       ILM = LM
 870   CONTINUE
        LQ = LN - (MOD(LN,465))
        LE1 =  IRY(LN)/MBIT + LQ
        LB  =  MOD(IRY(LN),MBIT)+1 +LQ
        LB1 = LB+10
        LE  =  LB1 +1
        IS =1
        IF (LCT.LT.LMAX) GO TO 880
        LPAGE = LPAGE +1
        LCT = 2
        IS = 2
 880    DO 890 J= 1,IOS
         IUT = OPU(J)
         GO TO (882,881),IS
 881     WRITE (IUT,9000) GREPID(1),GREPID(2),SPID,LPAGE
 882   P2(2)=P1(1)
       P2(5)=P1(2)
       P2(1)=P1(3)
       P2(4)=P1(4)
       P2(3)=P1(5)
       P2(6)=P1(6)
       WRITE(IUT,9004)(IRY(K),K=LB,LB1),(P2(K),K=KX,KXX)
 9004  FORMAT(11X,A6,5X,10A6,F7.2,2(F8.2),3X,3(F8.2))
         IF (LE.LT.LE1) WRITE (IUT,9005) (IRY(K),K=LE,LE1)
 9005    FORMAT ( 11X,A6,5X,10A6)
 890     CONTINUE
       LCT = (LE1-LB)/11 +1 +LCT
 900   CONTINUE
       REWIND IT
       REWIND ITMP
       TC(5) = GREPID(1)
       TC(7) = GREPID(2)
       IF(IOS.EQ.2) CALL REPEND(IAA1,SPID,TC,LSEQ)
       GO TO 400
 1000  CONTINUE
        REWIND IT
       CALL ERRMSG(3,'END OF GRPDIF')
       STOP
 901   FORMAT(A6,1X,A6,A2,1X,I1)
 902   FORMAT(13A6,A2)
 903   FORMAT(F5.0,1X,I1,1X,I1,1X,I1,1X,A6,1X,8A6)
 904   FORMAT(A4,9A6,18X,I4)
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      DIMENSION ARY(1000,13),NSUB(3,2),OPU(2),ISUB(6,3),HEAD(14,6),
     1          JA(50),KA(50),IRID(13),ISEQ(13),DUTY(2000),AKOS(2000),
     2          ANMP(2000),LTAB(2000),KTIT(500),SKP(1000),ITST(2),
     3         LAB(8),AA(50),JRID(13),IRY(1000,13)
      DIMENSION ISUB1(2),FDUTY(2000),KTAB(1000)
      DIMENSION SKI(13,8), ISKI(13,8), NMEMB(13)
      EQUIVALENCE (FDUTY,DUTY), (SKI,ISKI)
C
       EQUIVALENCE (OPU(1),IPR),
     2              (ARY(1,7),JA(1)),
     4              (END,LEND),
     5              (ARY(1,1),ANMP(1),DUTY(1)),
     6              (ARY(1,3),AKOS(1),LTAB(1)),
     7              (ARY(1,5),SKP (1))          ,(AA(50),JA(50)),
     8            (ARY(1,6),RA(1)),(ARY(1,1),IRY(1,1))
C
C
       INTEGER   RDR,OPU,DUTY,AA,SID1,SID2
      DATA ISUB1/12H DUTY  TASK /
      DATA NSUB/36HDUTY  DUTY  VAR IDDY-TSKDY-TSKVAR ID/
       DATA    KRD/6HGRPSUM/
C
C
       DATA    RDR/5/
       DATA    LEND/6HEND   /
      DATA ISUB/108HPERCENT MEMBERS PERFORMING          AVERAGE PERCENT
     1TIME BY ALL MEMBERS MEAN PROFILE ANALYSIS
       DATA    ITST/12HEND OF DATA /
       DATA    ILM/0/
      NCNT = 1000
      MBIT = 2**18
C
      CALL SETUP(10,27,26,0,0,0,20,0,0,0,0,JSEQ)
      IPT = 10
      IJD = 27
      IR = 26
      ITP = 20
      OPU(1) = 6
      OPU(2) = IR
      CALL DISK(250,465)
      IPR = 6
      ISKIP = 1
C
C      READ COMMUNICATION REGION AND CONTROL CARD
C
       READ (RDR,10) JRD,SID1,SID2
 10    FORMAT ( A6,1X,A6,A2)
       WRITE (IPR,50) JRD,SID1,SID2
 50    FORMAT ('1*** GROUP SUMMARY CONTROL CARDS ***',/,1X,A6,1X,A6,A2)
       READ  (IPT) K,(JA(I),I= 1,K)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          NTASK = JA(12)
          NDTY  = JA(9)
          NCASE = JA(3)
         NTKDY=NTASK+NDTY
          IF (JRD.EQ.KRD.AND.SID1.EQ.AA(1).AND.SID2.EQ.AA(2)) GO TO 100
          WRITE (IPR,51) KRD,JA(1),JA(2)
   51     FORMAT (45H0ERROR IN CONTROL CARD/TAPE,TAPE LISTED BELOW//,
         1            ,1X,A6,1X,A6,A2,/,12H0JOB ABORTED )
         STOP
  100     CONTINUE
C
C         READ HISTORY TAPE AND PLACE DUTY/TASK TITLES ON DISK.
C
          REWIND ITP
  110     READ (IPT) K,AKOS(1),AKOS(2)
          IF (LTAB(1).NE.ITST(1).AND.LTAB(2).NE.ITST(2)) GO TO 110
          IRCDS=0
  120     READ (IPT)  K,(LTAB(I),I=1,K)
          WRITE (ITP) K,(LTAB(I),I=1,K)
          IRCDS=IRCDS+1
          IF (LTAB(1).NE.ITST(1).AND.LTAB(2).NE.ITST(2)) GO TO 120
          IRCDS=IRCDS-3
          REWIND ITP
          CALL FREE(IPT)
          KTRK=1
          DO 150 IX=1,IRCDS
          READ (ITP) K,(LTAB(I),I=1,K)
          CALL DSKSTO (K,LTAB(1),KTRK,0)
          KTRK=KTRK+1
  150     CONTINUE
          READ(ITP) K,(KTAB(I),I=1,K)
          DO 155 I=1,NTKDY
          DUTY(I)=I
  155     CONTINUE
          REWIND ITP
         N = 1
         DO 160 LM = 1, NTKDY, NCNT
         LN = LM + NCNT - 1
         IF(LN.GT.NTKDY) LN = NTKDY
         K = LN - LM + 1
         WRITE(ITP) K, N, (ANMP(J),J=LM,LN)
  160    CONTINUE
C
C         MAIN LOOP PROCESSING
  200    ISKIP = 2
         READ(RDR,11,END=900) ITDY, KSW, NPG, IOS, SPID, LAB
   11    FORMAT(I1,1X,I1,1X,I1,1X,I1,1X,A6,1X,8A6)
         WRITE(IPR,13) ITDY, KSW, NPG, IOS, SPID, LAB
   13    FORMAT(1X,I1,1X,I1,1X,I1,1X,I1,1X,A6,1X,8A6)
         ISKIP = 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
         NCT = 50
         GO TO 210
 C
 C        PROGRAM ENDS HERE
 900     GO TO(901,205), ISKIP
 901     WRITE(IPR,902)
 902     FORMAT('1UNEXPECTED END OF FILE ON CARD READER')
         STOP
 205     CONTINUE
 C
         WRITE(IPR,211)
 211     FORMAT('1END OF GRPSUM')
         STOP
  210     CONTINUE
          NRQ  = 0
         IF(NPG.LE.6.AND.IOS.LE.1.AND.KSW.LE.2.AND.ITDY.LE.3) GO TO 207
         WRITE(IPR,541)
 541     FORMAT('0OPTIONS IN CC 1,3,5,OR 7 NOT IN RANGE. REPORT SKIPPED.')
         ISKIP = 2
 206     READ(RDR,542) (SKP(I),I=1,14)
         WRITE(IPR,7662) (SKP(I),I=1,14)
         IF(SKP(1).EQ.END) GO TO 200
         GO TO 206
 C
 207     IF(ITDY.EQ.0) ITDY = 2
         IF(KSW.EQ.0) KSW = 1
         IMAX = 12
         IF(KSW.EQ.2) IMAX = 6
         IF(IOS.NE.0) GO TO 801
         IOS = 2
 801     IF(NPG.NE.0) READ(RDR,542) ((HEAD(I,J),I=1,14),J=1,NPG)
 542     FORMAT(13A6,A2)
 C
 C        READ REQUEST CARDS
 C
          KSWT = 1
  300     NRQ = NRQ+1
  310    READ(RDR,12,END=389) SD,IRID(NRQ),(SKI(NRQ,I),I=1,8),ISEQ(NRQ)
   12    FORMAT(A4,9A6,18X,I4)
         IF(SD.EQ.END) GO TO 350
 C
          IF (NRQ.LE.IMAX) GO TO 300
          KSWT = 2
          GO TO 310
  350     CONTINUE
          NRQ = NRQ-1
 389     CONTINUE
          GO TO (352,351),KSWT
  351     WRITE (IPR,57)  IMAX
  57      FORMAT ('0MORE THAN',I3,' REQUEST CARDS IN ABOVE SET,EXTRA',
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
         1 ' CARDS SKIPPED.',/)
  352      CONTINUE
C         SEARCH FOR MAX SEQUENCE NUMBER
C
           MAX = 0
           DO 360 I= 1,NRQ
           IF(ISEQ(I).GT.MAX) MAX = ISEQ(I)
  360      CONTINUE
C
C          SEARCH BINARY DESCRIPTION TAPE FOR DESCRIPTIONS.
C
           ISQ = 0
           NR = 0
  400      IF (MAX.LE.ISQ ) GO TO 500
          CALL ZERO(AKOS,NTKDY)
          READ(IJD)IDSP,NT,(ANMP(K),K=1,NT),IK,(SKP(I),I=1,IK),ISQ,J,
         1          (   KA(K),K=1,J),(AKOS(K),K=1,NT),(AKOS(K),K=1,NT)
          DO 490 I=1,NRQ
          IF(ISEQ(I).NE.ISQ) GO TO 490
          KD = KA(35)
          IF(KD.NE.'JOBGRP'.AND.KD.NE.'JOBSPC'.AND.KD.NE.'SAMSEL') GO TO 1
          DO 2 K = 1, 8
          J = K + 36
          ISKI(I,K) = KA(J)
 2        CONTINUE
          IRID(I) = IDSP
 1        CONTINUE
          NMEMB(I)=KA(17)
          IF(ITDY.EQ.2.OR.NT.GE.NTKDY) GO TO 545
          WRITE(IPR,544)
   544 FORMAT(14H1***ILLEGAL***/ 90HODUTY GROUP SUMMARY REQUESTED FROM JO
         1B DESCRIPTION TAPE CONTAINING ONLY TASK INFORMATION. /
         216HOJOB TERMINATED.  )
          GO TO 205
   545 CONTINUE
              GO TO (450,470,470),KSW
  450         SD = KA(17)
C
          DO 460 LM=1,NT
          ANMP(LM)=ABS(AKOS(LM)/SD)*100.0
  460            CONTINUE
          GO TO 475
   470 DO 472 LM=1,NT
   472 ANMP(LM)=ABS(ANMP(LM))
   475 DO 480 LM=1,NTKDY,NCNT
              LN = LM +NCNT -1
          IF(LN.GT.NTKDY) LN=NTKDY
              K = LN- LM+1
           N = I+1
           WRITE (ITP) K,N,(ANMP(J),J=LM,LN)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 480      CONTINUE
 481    NR = NR +1
        ISEQ(I) =-ISEQ(I)
 490    CONTINUE
        GO TO 400
 500     CONTINUE
       NPAGE=1
       DO 1750 K=1,IOS
       IO=OPU(K)
       WRITE(IO,661) LAB, SPID, NPAGE
 661   FORMAT(1H1,8A6,29X,A6,6H PAGE ,I3,/)
       IF(NPG.NE.0) WRITE(IO,7662)((HEAD(L,M),L=1,14),M=1,NPG)
 7662 FORMAT(1X,13A6,A2)
       WRITE(IO,8662)
 8662 FORMAT(1H0,46HREPORTS ON THE FOLLOWING GROUPS WERE REQUESTED  //)
       DO 1950 KK=1,NRQ
 1950 WRITE(IO,1951) IRID(KK),(SKI(KK,M),M=1,8),NMEMB(KK)
 1951 FORMAT(1H ,8X,18HGROUP IDENTITY =  A6,2X,8A6,1X,10HCONTAINING,I6,
     18H MEMBERS.  )
 1750 CONTINUE
        REWIND ITP
 502    REWIND IJD
C
        IF (NR.EQ.NRQ) GO TO 540
        DO 510 I = 1,NRQ
        IF (ISEQ(I).GT.0) WRITE (IPR,58) ISEQ(I)
 58     FORMAT (34H0CANNOT FIND REPORT WITH SEQUENCE=,I4 )
 510    CONTINUE
        WRITE (IPR,60)
 60     FORMAT (38H0THE ABOVE REPORT HAS BEEN SKIPPED.      /)
        GO TO 200
 540    CONTINUE
        NQ = NRQ +1
C
C     SPLIT REPORT ID AND PRINT 2 LINES SUCH AS SPC  GP    GP
C                                               001  1000  0010
C THIS SECTION MUST BE RECODED * * * *
        DO 745 IX=1,NRQ
       IF(KSW.EQ.2) GO TO 745
       DECODE(741,IRID(IX)) ID1, ID2
       CALL BCDBIN(ID2,K,4,$737)
       GO TO 739
 737   DECODE(738,IRID(IX)) ID1, ID2
 739   IRID(IX) = ID1
       JRID(IX) = ID2
 738    FORMAT(A3,A3)
 741    FORMAT(A2,A4)
 745  CONTINUE
C       PRINT DETAIL REPORT,READING DATA FROM TAPE
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      K1=2-MOD(ITDY,2)
      K2=(ITDY+2)/2
      DO 800 KOP=K1,K2
      LCT=MBIT
C
C
       NT = 0
      ML=NTKDY/NCNT
      IF(MOD(NTKDY,NCNT).NE.0) ML=ML+1
      M1=1
      IF(KOP.EQ.1) M1=ML
      DO 790 I=M1,ML
         DO 730  K = 1,NQ
            DO 710 J=1,I
        READ (ITP) L,N,(ARY(M,N) ,M=1,L)
            MJ =J
 710        CONTINUE
            IF (MJ.EQ.ML) GO TO 730
            MJ = MJ + 1
            DO 720 L=MJ,ML
            READ (ITP) M
 720        CONTINUE
 730     CONTINUE
       REWIND ITP
      IB=1
      IE=NCNT
      IF(KOP.EQ.2) GO TO 731
      IB=NTASK+1
      IE=NTKDY
  731 CONTINUE
C
      DO 781 J=IB,IE
         NT = NT +1
         IF (NT.GT.NTASK) GO TO 800
         IF (LCT.LT.NCT) GO TO 751
C
C       PRINT PAGE HEADINGS
C
         NPAGE = NPAGE +1
            LCT = 3
            DO 750 K =1,IOS
            IO = OPU(K)
      WRITE(IO,61) LAB, SPID, NPAGE, ISUB1(KOP)
 61   FORMAT(1H1,8A6,29X,A6,6H PAGE ,I3,//,1X,A6,' GROUP SUMMARY')
      WRITE(IO,6161) (ISUB(L,KSW),L=1,6)
 6161 FORMAT(2X,6A6)
            IF (NPAGE.NE.1) GO TO 746
        WRITE (IO,62)
            IF (NPG.NE.0) WRITE (IO,62) ((HEAD(L,M),L=1,14),M=1,NPG)
 62   FORMAT(1X,13A6,A2)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
                IF (K.EQ.1) LCT =LCT+ NPG+1
  746  IF(KSW.NE.2) GO TO 1746
       WRITE(IO,1364) NSUB(KSW,KOP),(IRID(L),L=1,NRQ)
 1364  FORMAT('0',/,28X,A6,34X,6(4X,A6))
       GO TO 1365
 1746   WRITE(IO,63) (IRID(L),L=1,NRQ)
   63   FORMAT('0',67X,12(A3,2X))
        WRITE(IO,64) NSUB(KSW,KOP),(JRID(L),L=1,NRQ)
   64   FORMAT(' ',27X,A6,34X,12(A4,X) )
 1365  WRITE(IO,67)
   67  FORMAT(' ')
  750        CONTINUE
  751      CONTINUE
C
C       PRINT REPORT DETAILS
C
             DO 780 K =1,IOS
             IO = OPU(K)
        IF(K.NE.1) GO TO 762
        DO 760 M=2,NQ
        IF(KSW.NE.2)  IRY(J,M)=ARY(J,M)+.5
  760   CONTINUE
C TASK TITLE RETREIVAL AND PRINTING SECTION
  762  NN=IRY(J,1)
        KTRK=KTAB(NN)/MBIT
        ILOC=MOD(KTAB(NN),MBIT)
        CALL DSKRED(465,KTIT(1),KTRK,0)
        ISTART=ILOC+1
        ISTOP=ISTART+10
        IRCDS=MOD(KTIT(ILOC),MBIT)
        IRCDS= (KTIT(ILOC)/MBIT - IRCDS) / 11
        ILINES=IRCDS
       IF(KSW.NE.2)
      1 WRITE(IO,65)(KTIT(IX),IX=ISTART,ISTOP),(IRY(J,M),M=2,NQ)
       IF(KSW.EQ.2)
      1 WRITE(IO,965)(KTIT(IX),IX=ISTART,ISTOP),(ARY(J,M),M=2,NQ)
   65  FORMAT(' ',11A6,1X,12(I3,2X))
  965  FORMAT(' ',11A6,1X,9F10.4)
  761   IF(IRCDS.EQ.1) GO TO 780
        IRCDS=IRCDS-1
        ISTART=ISTART+11
        ISTOP=ISTOP+11
        WRITE(IO,66)(KTIT(IX),IX=ISTART,ISTOP)
   66   FORMAT(' ',11A6)
C END OF SECTION
  780   CONTINUE
        LCT=LCT+ILINES
  781     CONTINUE
  790   CONTINUE
  800   CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      IF(IOS.EQ.2) CALL REPEND(SIDI,SPID,LAB,JSEQ)
       WRITE (IPR,50)
       READ (ITP) M
      GO TO 200
       END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      COMMON/FGEN/IFMT(1000),ITSKTB(1700),IHISTB(1700),ICHKTB(500),
     1ICHEK(500), LFWTB( 50), NTHTB( 50), NCHEK, NCARD, KTM, KTT
      DIMENSION ID(2),IBUF(4000),COMM(50),RELLOC(2000),TIDT(100),
     1 HDICT(15,17), HVARS(2000), TVARS(2000),
     2    IDTCR(99),IFMAT(40),FWATE(200),  TLOC(2099),NTID(99),JBUF(99),
     3KBUF(11),ABUF(4000),CDIC1(105),CDIC2(45),CDICT(150)
      DIMENSION INFILE(1600), MBUF(14)
      DOUBLE PRECISION ISM10,ISQ10
      EQUIVALENCE (IBUF,COMM),          (IBUF(51),RELLOC),
     1              (IBUF(2186),HDICT),
     2             (IBUF(3200),JBUF),    (IBUF(1000), TLOC),
     3             (IBUF,ABUF),          (CDICT(1),CDIC1(1)),
     4             (CDICT(106),CDIC2(1))
      INTEGER COMM,RELLOC,CDICT,HDICT,HVARS,TVARS,TLOC,TIDT
      DATA KON1,KON2,KON3/1H ,07777777777777,03777777777777/
      DATA (INFILE(I),I=1,11)/'10',6*' ',751,14,'IDSRLP',' '/
      MOV2=2**12
      MOV3=2**18
      MOV4=2**24
      MOV5=2**30
      ISM10 = 0.0
      ISQ10 = 0.0
      NCV=10
      NO = 0
       INP=10
       IHT=15
       IO=20
       CALL SETUP(10,15,0,0,0,0,20,0,0,0,0,JSEQ)
C
C***** IHT - HISTORY TAPE
C***** ID -STUDY INDENTIFICATION
C***** NCASE - NO. OF CASES
C***** NTASK - NO. OF TASKS PER CASE
C***** NDUTY - NO. OF DUTIES
C***** NCARD - NO. OF CARDS PER CASE
C***** MXTSK - MAX VALUE FOR TASKS
C***** NHIST - NO. OF HISTORY VARIABLES PER CASE
C***** KOPT - SET TO 1 WHEN WEIGHTS AND SUBSTITUTION OPTION IS USED
C***** READ CONTROL CARD
      READ 900,MXTSK,NCARD,NTASK,NHIST,NDUTY,      ID(1),ID(2),NCASE,
     1KOPT
  900 FORMAT(6X,I6,I3,I5,I5,I3, X,     A6,A2,I6,I2)
C     READ WEIGHTS IF REQUIRED
      IF(KOPT.EQ.0) GO TO 47
      GO TO(35,35,45,45),KOPT
   35 READ 950, FWATE(1)
  950 FORMAT(F8.0)
      GO TO (36,47),KOPT
   36 FMXTSK=MXTSK-1
      EXP=1./FMXTSK
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      EXP=FWATE(1)**EXP
      EXPON=0.
      DO 37 I=1,MXTSK
      FWATE(I)=EXP**EXPON
   37 EXPON=EXPON+1.
   45 READ 951,(FWATE(I),I=1,MXTSK)
  951 FORMAT(13F6.0)
C
C***** GENERATE FORMAT AND SET UP INPUT TABLES
C***** KTT   - COUNT OF THE NUMBER OF TASK VARIABLES
C***** KTH   - COUNT OF THE NUMBER OF HISTORY VARIABLES
   47 LOPT=1
      CALL FMTGEN(LOPT)
      IF(NTASK.EQ.KTT) GO TO 50
      PRINT 953, NTASK,KTT
  953 FORMAT('INTASK FROM CONTROL CARD IS ',I4,'.  THE T COUNT FROM FORM
     1AT CARDS IS ',I4,'.  RUN ABORTED.')
      STOP
   50 IF(NHIST.EQ.KTH) GO TO 180
      PRINT 955, NHIST,KTH
  955 FORMAT('INHIST FROM CONTROL CARD IS ',I4,'.  THE H COUNT FROM FORM
     1AT CARDS IS ',I4,'.  RUN ABORTED.')
      STOP
C
C***** STORE VALUES IN COMMUNICATION REGION
C***** WRITE COMMUNICATION REGION ON TAPE
C***** COMM - COMMUNICATION REGION
  180 DO 190 I=1,50
  190 COMM(I)=0
      COMM(1)=ID(1)
      COMM(2)=ID(2)
      COMM(3)=NCASE
      COMM(4)=(NHIST+15)/16+1
      COMM(6) = ' '
      COMM(9)=NDUTY
      COMM(10)=NHIST
      COMM(11)=NCV
      COMM(12)=NTASK
      N=50
      WRITE(IMT) N,(COMM(I),I=1,N )
C
C***** THIS SECTION FORMS AND WRITES THE RELATIVE LOCATION TABLE,
C***** TASKS IN DUTY TABLE, COMPUTED VARIABLE DICTIONARY, AND HISTORY
C***** VARIABLE DICTIONARY.
C***** RELLOC- HISTORY VAR. LOC. TABLE
C***** HDICT  - HIST. VAR. DICT.
C***** CDICT - COMPUTED VARIABLE DICTIONARY
C*****  TIDT - TASK IN DUTY TABLE
C***** IDTCR - ARRAY CONTAINING DUTY CHAR.
C***** NTID - ARRAY CONTAINING NO. TASKS PER DUTY
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C***** FORMAT TYPE 1 - MA6,1AN
C***** FORMAT TYPE 2 - MA6
C***** FORMAT TYPE 3 - 1AN
C***** M1 - FIRST CHAR OF M
C***** M2 - SECOND CHAR OF M
C***** IW - TEMP STORAGE FOR FIRST WORD OF FORMAT
C***** KTH - COUNTS HIS. VAR.
C***** KTHC - COUNTS HIS. VAR. IN CURRENT GROUP OF 16
C***** L - LENGTH OF HIS. VAR. (CHAR.)
C***** NDICT COUNTS NO. OF DICTIONARY GROUPS
       DATA CDIC1/630HC001 = CASE IDENTIFICATION ASSIGNED BY INPSTD
      1                                     14C002 = NUMBER OF NON-ZERO R
      2ESPONSES TO THE QUESTIONAIRE
      33 = KPATH SEQUENCE NUMBER (INTERNAL USE ONLY)                14C00
      4                        14C004 = DECISION VALUE USED TO COMBINE CASES A
      5T THIS STAGE (BEST OR AVERAGE BETWEEN)    F7.2C005 = AVERAGE OF ALL
      6 VALUES IN THIS GROUP AT A GIVEN STAGE (AVERAGE WITHIN)            F
      77.2C006 = W(I), WEIGHT ASSOCIATED WITH THE I-TH INPUT CASE
      8                          F6.0C007 = NUMBER OF NON-ZERO RESPONSES TO
      9THE QUESTIONAIRE                              F5.0/
       DATA CDIC2 /180HC008 = SUM OF RAW TASK RESPONSES
      1                                     F6.0C009 = KPATH SEQUENCE NUMB
      2ER                                                             F6.0,
      3'C010 = NUMBER OF NON-ZERO RESPONSES TO THE QUESTIONAIRE SQUARED',
      4 3*' ',' F11.0'/
C***** READ DUTY CHAR. AND NO. TASKS IN DUTIES
      DO 192 I=1,NDUTY
      READ 977, IDTCR(I),NTID(I)
  977 FORMAT(3X,A1,I4)
  192 CONTINUE
C***** FORM TASK IN DUTY TABLE
C***** IE - TASK NO. THAT ENDS A DUTY
C***** IB - TASK NO. THAT BEGINS A DUTY
      IE=0
      DO 195 I=1,NDUTY
      IB=IE+1
      IE=IE+NTID(I)
  195 TIDT(I)=IB*MOV3+IE
C***** FORM RELATIVE LOCATION TABLE (RELLOC)
      LOC=1
      DO 197 I=1,NHIST
      NC=IHISTB(I)/MOV3
      NW=(NC+5)/6
      RELLOC(I)=LOC+MOV3*NW
  197 LOC=LOC+NW
      NHW=LOC-1
C***** WRITE HIS. VAR. LOC. TABLE, TASK IN DUTY TABLE ON TAPE
      WRITE(IHT)NHIST,(RELLOC(I),I=1,NHIST),NDUTY,(TIDT(I),I=1,NDUTY)
C***** WRITE COMPUTED VARIABLE TITLES
      LENCD = 15*NCV
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      WRITE(IHT)NCV,(CDICT(I),I=1,LENCD)
      KTH=1
      NDICT=1
C***** LOOP FOR GROUP OF 17 HISTORY VARIABLES
  200 KTHC=1
C***** LOOP FOR EACH HISTORY VARIABLE - READ HISTORY TITLE
  205 READ 978,(HDICT(I,KTHC),I=1,14)
  978 FORMAT(13A6,A2)
      L=IHISTB(KTH)/MOV3
      IF(L.GT.6) GO TO 208
      M = 1
      N = L
  202 ENCODE(600,HDICT(15,KTHC)) M, N
  600  FORMAT(I4,1HA,I1)
       GO TO 220
  208 M = L/6
      N = L - 6*M
      IF(N.NE.0) GO TO 210
      N = 6
      GO TO 202
  210  ENCODE(601,HDICT(14,KTHC)) M,N
  601  FORMAT(T5,I2,5HA6,1A,I1)
  220  CONTINUE
C***** CHECK IF LAST TITLE HAS BEEN READ
      IF(KTH.GE.NHIST) GO TO 245
      KTH=KTH+1
C***** CHECK FOR 16 TITLES IN THE CURRENT GROUP
      IF(KTHC.GE.16) GO TO 240
      KTHC=KTHC+1
      GO TO 205
  240 NDICT=NDICT+1
C***** WRITE THE CURRENT GROUP OF TITLES ON TAPE
      N16=16
      WRITE(IHT)N16,((HDICT(I,J),I=1,15),J=1,16)
      GO TO 200
C***** WRITE FINAL GROUP ON TAPE
  245 WRITE(IHT)KTHC,((HDICT(I,J),I=1,15),J=1,KTHC)
C
C***** THIS SECTION READS ONE TASK OR DUTY TITLE
C***** JBUF - STORAGE AREA FOR TITLE
C***** NW - NO. OF WORDS IN TITLE
C***** IEND - END OF DATA INDICATOR
C***** IDTCR - ARRAY OF DUTY CHAR.
C***** NTID - ARRAY OF NO. OF TASKS IN DUTIES
C***** NTASK - NO. OF TASKS
C***** NDUTY - NO. OF DUTIES
C***** NTR - NO. OF TITLES READ THUS FAR
C***** KBUF - STORAGE FOR TITLE ON LOOK-AHEAD CARD
C***** IFIRST - FIRST PASS INDICATOR
C***** KEND - END OF DATA IND. FOR LOOK-AHEAD CARD
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
C***** KTD - COUNTS DUTIES
C***** KTTD - COUNTS TASKS IN DUTY
C***** IDUT - DUTY CHAR. FOR CURRENT TASK
C***** KDUT - DUTY CHAR. ON LOOK-AHEAD CARD
C***** ISEQ - SEQUENCE NO. ON CURRENT CARD
C***** JSEQ - SEQUENCE NO. ON LOOK-AHEAD CARD
       DATA IBLANK, IEND3/' ','END '/
       K=0
       NTR=0
       IL=0
       NGT=1
       IEND=0
       IQUIT=0
       IFIRST=1
       KEND=0
       KTD=1
       KTTD=0
C***** TEST FOR FIRST PASS
  250  IF(IFIRST.EQ.1) GO TO 270
C***** TEST FOR END OF DATA ON LOOK-AHEAD CARD
       IF(KEND.NE.1) GO TO 255
       IEND=1
       IF(IQUIT.EQ.0) GO TO 315
       PRINT 275
  275  FORMAT('1ERROR IN DUTY/TASK TITLE CARDS.  ABORTED.')
       STOP
  255  NW=0
C***** MOVE LOOK-AHEAD TITLE INTO CURRENT TITLE
  260  DO 265 I=1,11
       J=NW+I
  265  JBUF(J)=KBUF(I)
       NW=NW+11
       CALL ZERBLK(JBUF,IDUT)
       ISEQ=JSEQ
       GO TO 295
C***** FIRST PASS - READ INITIAL CARD
  270  IFIRST=0
       NW=11
  280  READ    979, ITEST,ISEQ,(JBUF(I),I=1,11)
  979  FORMAT(A4,A1,1X, 11A6)
       CALL BCDBIN(ISEQ,ISEQ,1,$285)
C***** PICK OFF DUTY CHAR.
  285  CONTINUE
       CALL ZERBLK(JBUF,IDUT)
  286  FORMAT(A1)
C***** READ LOOK-AHEAD CARD
  295  READ(5,979,END=299) ITEST,JSEQ,(KBUF(I),I=1,11)
       CALL BCDBIN(JSEQ,JSEQ,1,$297)
C***** PICK OFF DUTY CHAR.
  297  DECODE(286,KBUF(1)) KDUT
```

LINE # TEXT...

```
      GO TO 300
C***** EOF ON CARD READER
 299  KEND=1
      GO TO 305
C***** CHECK FOR NEW TITLE
 300  IF(JSEQ.EQ.1) GO TO 305
      IF(JSEQ.NE.ISEQ+1) PRINT 980, NTR
      IF(JSEQ.NE.ISEQ+1)IQUIT=1
  980 FORMAT(49H  A CARD IS OUT OF SEQUENCE IN TASK TITLE NUMBER I4,2H.
     1)
      KBUF(1)=IBLANK
      GO TO 260
 305  IF(KTD.GT.NDUTY) GO TO 315
C***** CHECK FOR NEW DUTY CHAR.
      IF(IDUT.NE.IDTCR(KTD)) GO TO 310
      KTTD=KTTD+1
      GO TO 315
C***** TEST NO. OF TASKS IN DUTY
  310 IF(KTTD.NE.NTID(KTD))WRITE(6,981)IDTCR(KTD),NTID(KTD),KTTD
      IF(KTTD.NE.NTID(KTD))IQUIT=1
  981 FORMAT(11H  FOR DUTY A1,I5,31H TASK TITLES WERE EXPECTED, AND I5,
     1 22H TITLES WERE PRESENT. )
      KTD=KTD+1
      KTTD=1
C
C***** THIS SECTION WRITES TASK TITLES,    DUTY-TASK-LOCATOR-TABLE, AND
C*****     TASK-IN-DUTY-TABLE ON TAPE
C***** NW - NO. OF WORDS IN THE CURRENT TITLE
C***** IL - INDEX OF LAST WORD IN THE PREVIOUS TITLE
C***** K - NO. OF WORDS IN THE CURRENT RECORD
C***** NGT - NO. OF GROUPS OF TITLES
C***** NTR - NO. OF TITLES READ THUS FAR
C***** TLOC  - DUTY TASK LOCATOR TABLE
C***** IBUF - STORAGE AREA FOR A GROUP OF TITLES
C***** JBUF - STORAGE AREA FOR ONE TITLE
C***** IEND - END OF DATA INDICATOR
C***** INITIALIZE COUNTERS
C***** TEST FOR END OF DATA
 315  IF(IEND.EQ.1) GO TO 330
      NTR=NTR+1
      K=K+1+NW
      IF(K.LE.465) GO TO 320
      K=K-1-NW
      NGT=NGT+1
C***** WRITE GROUP OF TITLES ON TAPE
      WRITE(10 )K,(IBUF(I),I=1,K)
      K=NW+1
      IL=0
C***** STORE TITLE IN GROUP
 320  DO 325 I=1,NW
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      J=IL+I+1
  325 IBUF(J)=JBUF(I)
      IBUF(IL+1)=(IL+1+NW)*MOV3+(IL+1)
C***** MAKE ENTRY IN DUTY TASK LOCATOR TABLE
      TLOC(NTR)=MOV3*NGT+IL+1
      IL=K
      GO TO 250
C***** WRITE FINAL GROUP OF TITLES ON TAPE
  330 WRITE(IO ) K,(IBUF(I),I=1,K)
C***** WRITE DUTY TASK LOCATOR TABLE ON TAPE
      NDT=NDUTY+NTASK
      WRITE(IO)NDT,(TLOC(I),I=1,NDT)
C***** WRITE TASK IN DUTY TABLE ON TAPE
      TIDT(NDUTY+1)=(NGT+1)*MOV3
      ND1=NDUTY+1
      WRITE(IO)ND1,(TIDT(I),I=1,ND1)
C***** WRITE END OF DATA RECORD ON TAPE
      N2=2
      WRITE(IO )N2,IEND1,IEND2
      REWIND IO
      IF(NDT.EQ.NTR) GO TO 335
      PRINT 982, NDT,NTR
  982 FORMAT('1DUTY/TASK TITLES EXPECTED ',I4,'.  TITLES READ ',I4,
     1'.  RUN ABORTED.')
      STOP
C
C***** THIS SECTION READS AND EDITS CASE DATA, CONVERTS TASK VALUES TO
C*****     PERCENTAGES, AND WRITES CASE DATA ON TAPE
C***** IBUF - STORAGE AREA FOR INPUT DATA
C***** IDA - ASSIGNED CASE ID (SEQUENTIAL)
C***** ISUM - SUM OF TASK VALUES
C***** IVAL - TASK VALUE
C***** LNTH - LENGTH OF A HISTORY VARIABLE
C***** LOC - LOCATION TO START STORING A HISTORY VARIABLE
C***** NNZ - NO. OF NON-ZERO VALUES IN A CASE
C***** NCHEK - NO. OF CHECK COLUMNS PER CASE
C***** NTH - NO. THINGS TO READ INTO IBUF EACH CASE
C***** NHW - NO. HISTORY WORDS
C***** IHISTB - TABLE CONTAINING LENGTH(CHAR) AND POSITION OF HIS VARS
C***** ITSKTB - TABLE CONTAINING POSITION OF TASK VARIABLES
C***** ICHKTB - TABLE CONTAINING POSITION OF CHECK VARIABLES
C***** ICHEK  - TABLE CONTAINING CHECK VARIABLES
C***** IFMT   - GENERATED FORTRAN FORMAT
C***** LFWTB  - TABLE CONTAINING POSITION OF END FORMAT FOR EACH CARD
C***** NTHTB - TABLE CONTAINING NUMBER OF THING TO READ PER CARD
      DATA IEND1/6HEND OF/,IEND2/6H DATA /
  335 KCASE=0
      IPREV=0
      F1=1.
      IDA=0
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      IB=1
C***** LOOP FOR EACH CASE - READ CASE DATA INTO IBUF
  340 LAST=0
      NTH=0
      DO 365 I=1,NCARD
C***** PICK UP THE FORMAT FOR THIS CARD
      LAS1=LAST+1
      LAST=LFMTB(I)
      DO 345 J=LAS1,LAST
      K=J-LAS1+1
  345 IFMAT(K)=IFMT(J)
C***** READ ONE CARD
      NTH1=NTH+1
      NTH=NTHTB(I)
      CALL READF(INFILE,MBUF,$501)
      DECODE(IFMAT,MBUF) (IBUF(J),J=NTH1,NTH)
       IF(IBUF(NTH1).EQ.IENDJ) GO TO 501
  365 CONTINUE
      JJW = FLD(0,24,IBUF(1))
      IF(JJW.GE.IPREV) GO TO 367
      PRINT 984,IPREV,IBUF(1)
  984 FORMAT('1CASE DATA OUT OF SEQUENCE.',' CASE ',A4,' FOLLOWED BY ',
     1A4,'. RUN ABORTED.')
      STOP
  367 IPREV=JJW
C***** CHECK COLUMNS THAT ARE TO MATCH FORMAT CHARS FOR FIRST CASE ONLY
      IF(NCHEK.EQ.0) GO TO 375
      DO 370 I=1,NCHEK
      J=ICHKTB(I)
      IF(ICHEK(I).EQ.IBUF(J)) GO TO 370
      PRINT 987, ICHEK(I),IBUF(J)
  987 FORMAT('1CHECK CHARACTER NO MATCH.  FORMAT CARD CHARACTER IS ',A1,
     1'.  DATA CHARACTER IS ',A1,'.  RUN ABORTED.')
       STOP
  370 CONTINUE
      NCHEK=0
C***** CHECK TO SEE IF WEIGHT AND SUBSTITUTION OPTION IS BEING REQUESTED
  375 IF(KOPT.NE.0) GO TO 425
C***** NO OPTION - CONVERT RESPONSES TO PERCENTS
      ISUM=0
      NNZ = 0
      DO 395 I=1,NTASK
      J=ITSKTB(I)
      IF(IBUF(J).NE.0)NNZ=NNZ+1
      ISUM=ISUM+IBUF(J)
  395 CONTINUE
      K=1
      DO 415 I=1,NTASK
      J=ITSKTB(I)
      IF(IBUF(J).EQ.0) GO TO 415
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      TVARS(K)=MOV3*I+IBUF(J)*100000/ISUM
      K=K+1
  415 CONTINUE
      SM=ISUM
      GO TO 485
C***** WEIGHT AND SUBSTITUTION OPTION.  CHANGE RESPONSES, THEN
C***** CONVERT THEM TO PERCENTS
  425 NNZ=0
      SM=0.
      DO 480 I=1,NTASK
      J=ITSKTB(I)
      IF(IBUF(J).EQ.0) GO TO 480
      GO TO (455,460,455,465),KOPT
  455 IZ=IBUF(J)
      ABUF(J)=FWATE(IZ)
      GO TO 470
  460 FBUF=IBUF(J)
      ABUF(J)=FBUF**FWATE(I)
      GO TO 470
  465 FBUF=IBUF(J)
      IZ=IBUF(J)
      ABUF(J)=FBUF*FWATE(IZ)
  470 IF(ABUF(J).EQ.0.) GO TO 480
      SM=SM+ABUF(J)
      NNZ=NNZ+1
  480 CONTINUE
      IF(NNZ.NE.0) GO TO 481
      PRINT 988, IBUF(1)
  988 FORMAT('1THE WEIGHT AND SUBSTITUTION OPTION HAS RESET ALL RESPONSE
     1S FOR CASE ',A4,' TO ZERO.  NOT ALLOWED.  RUN ABORTED.')
      STOP
C***** CONVERT TASK VALUES TO PERCENTAGES
  481 K=1
      DO 482 I=1,NTASK
      J=ITSKTB(I)
      IF(ABUF(J).EQ.0.) GO TO 482
      IVAL=ABUF(J)*100000./SM
      TVARS(K)=MOV3*I+IVAL
      K=K+1
  482 CONTINUE
C***** STORE HISTORY DATA
  485 LOC=0
      DO 495 I=1,NHIST
      LNTH=IHISTB(I)/MOV3
      LNTH=(LNTH+5)/6
      J=MOD(IHISTB(I ),MOV3)
      DO 491 K=1,LNTH
      L=LOC+K
      M=J+K-1
  491 HVARS(L)=IBUF(M)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

INPSTD INPUT STANDARD PROGRAM

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
  495 LOC=LOC+LNTH
C***** WRITE CASE DATA ON TAPE
      FNZ=NNZ
      FNZSQ = FNZ*FNZ
      ISM10 = ISM10 + FNZSQ
      ISQ10 = ISQ10 + FNZSQ*FNZSQ
      IDA=IDA+1
      WRITE (IHT) NCV,IDA,NNZ,NO,NO,NO,F1,FNZ,SM,NO,FNZSQ,NHW,
     1(HVARS(I),I=1,NHW),NNZ,(TVARS(I),I=1,NNZ)
      CALL STATS(NNZ)
      KCASE=KCASE+1
      GO TO 340
  501 IF(KCASE.EQ.NCASE) GO TO 502
      PRINT 990,NCASE,KCASE
  990 FORMAT('1CASES EXPECTED ',I4,'.  CASES ENCOUNTERED ',I4,'.  RUN AB
     1ORTED.')
      STOP
C***** WRITE END OF DATA RECORD ON TAPE
  502 N1=1
      N2=2
      WRITE(IHT)N2,IEND1,IEND2,N1,NO,N1,NO
  505 READ(IO) K,(IBUF(I),I=1,K)
      WRITE(IHT) K,(IBUF(I),I=1,K)
      IF(IBUF(1).NE.IEND1.OR.IBUF(2).NE.IEND2)GO TO 505
      ENDFILE IHT
      PRINT 998, ID(1),ID(2),NCASE,NTASK,NDUTY,NHIST,NCV
  998 FORMAT('1INPSTD COMPLETE - HISTORY TAPE CONTAINS...',///,
     1' ',A6,A2,'    STUDY ID',/,
     2' ',I8   ,'    CASES',/,
     3' ',I8   ,'    TASKS',/,
     4' ',I8   ,'    DUTIES',/,
     5' ',I8   ,'    HISTORY VARIABLES',/,
     6' ',I8   ,'    COMPUTED VARIABLES')
      CALL STATSS(AMEAN,SD,MINT,MAXT,ISUM,KOUNT)
      X = ISUM
      Y = NTASK*NCASE
      DENSTY = X/Y*100.0
      Z = NCASE
      XBAR = ISM10/Z
      SD10 = SQRT(ABS(ISQ10/Z - XBAR*XBAR))
      WRITE(6,888) AMEAN,SD,XBAR,SD10,MINT,MAXT,ISUM,DENSTY
      CALL RUNID(IDRUN,ISTUDY)
      CALL ERTRAN(9,KDATE,KTIME)
      WRITE(1,857) IDRUN,ISTUDY,ID(1),ID(2),AMEAN,SD,MINT,MAXT,NTASK,
     1 ISUM,DENSTY,KOUNT,KDATE
  857 FORMAT(A4,1X,A4,1X,A6,A2,1X,F9.3,1X,F9.4,1X,I4,1X,I4,1X,I4,
     1 1X,I8,1X,F6.2,1X,I5,1X,A4)
  888 FORMAT(///,' TASK SUMMARY INFORMATION',//,15X,'C007',/,1X,F11.2,
     1 ' = MEAN NUMBER OF TASKS PERFORMED',/,1X,F11.4,' = ',
     2 'STANDARD DEVIATION OF NUMBER OF TASKS PERFORMED',//,15X,'C010',
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
     A /,1X,F11.0,' = MEAN OF NUMBER OF TASKS PERFORMED SQUARED',/,1X,
     B F11.3,' = STANDARD DEVIATION OF TASKS PERFORMED SQUARED',//,1X,
     3 I11,' = MINIMUM NUMBER OF TASKS PERFORMED BY AN INDIVID',
     4 'UAL',/,1X,I11,' = MAXIMUM NUMBER OF TASKS PERFORMED BY',
     5 ' AN INDIVIDUAL',/,1X,I11,' = TOTAL NUMBER OF TASK RESPONSES',/,
     6 1X,F11.2,' = PERCENTAGE OF TOTAL POSSIBLE RESPONSES')
      WRITE(6,999)
  999 FORMAT('1PLEASE RETURN PUNCHED CARD TO SMAZ PROGRAMMERS (F-8)')
      STOP
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
$(0)      AXR$      .
ASAVE     RES       1      .
PCT       RES       24                 .
          . THIS FORTRAN CALLABLE SUBROUTINE WILL RETURN THE RUNID AND
          . STUDY (ACCOUNT#) FROM THE PROGRAM CONTROL TABLE(PCT).
          . THE CALLING SEQUENCE IS:
          .     CALL RUNID(IDRUN,ISTUDY)
          .
$(1)      ERU$      .
RUNID*    SA        A0,ASAVE           .
          LA        A0,(24,PCT)        .
          ER        PCT$               .
          LA        A0,PCT             .
          SA        A0,*0,X11          .
          LA        A0,PCT+023         .
          SA        A0,*1,X11          .
          LA        A0,ASAVE           .
          J         3,X11              .
          END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE ZERBLK(ID,IDUT)
      DIMENSION ID(11)
      DECODE(100,ID(1)) IDUT, ITASK
100   FORMAT(A1,A4)
      ID2 = ID(1)
      IF(ITASK.EQ.' ') RETURN
      CALL BCDBIN(ITASK,JTASK,4,$9)
      ENCODE(101,ID(1)) IDUT, JTASK, ID2
101   FORMAT(A1,I4,R1)
      RETURN
9     CONTINUE
      WRITE(6,102) ID
102   FORMAT(1X,A6,' IS INVALID TASK NUMBER.  RUN HALTS.')
      CALL ERTRAN(2)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      DIMENSION BYPASS(2000), JA(50), INPUT(3), ITITLE(8), ID(3)
      EQUIVALENCE (JA(6),MATRIX), (JA(17),MEMBRS), (JA(37), ITITLE(1))
      COMMON/INFILE/ IDPROG, INPUT
      INTEGER BLANK
      DATA BLANK, ID /' ', 'ON MAGNETIC TAPE'/
      CALL SETUP(27,0,0,0,0,0,0,0,0,0,0,JSEQ)
      IF(INPUT(1).NE.0) GO TO 4
      DO 3 I = 1, 3
      INPUT(I) = ID(I)
    3 CONTINUE
    4 CONTINUE
      INP = 27
      REWIND INP
      READ(INP,ERR=777,END=998) IDREPT,NTD,(BYPASS(I),I=1,NTD),NC35
     1,(BYPASS(I),I=1,NC35),JSEQ,NJA,(JA(I),I=1,NJA)
      ID1 = JA(1)
      ID2 = JA(2)
      NCASE = JA(3)
      NTASK = JA(12)
      NDUTY = JA(9)
      IPAGE = 0
      REWIND INP
    1 IPAGE = IPAGE + 1
      WRITE(6,100) INPUT, IPAGE, ID1, ID2, NCASE, NTASK, NDUTY
      DO 5 NREPT = 1, 49
      READ(INP,ERR=777,END=999) IDREPT,NTD,(BYPASS(I),I=1,NTD),NC35
     1,(BYPASS(I),I=1,NC35),JSEQ,NJA,(JA(I),I=1,NJA)
      IF(MATRIX.NE.BLANK.AND.MATRIX.NE.0) GO TO 2
      WRITE(6,101) ITITLE, IDREPT, MEMBRS, JSEQ
      GO TO 5
    2 WRITE(6,102) ITITLE, IDREPT, MATRIX, MEMBRS, JSEQ
    5 CONTINUE
      GO TO 1
  777 WRITE(6,103)
      STOP
  998 WRITE(6,105)
      STOP
  999 WRITE(6,104)
      STOP
  100 FORMAT('1',41X,'INDEX OF JOB DESCRIPTION FILE ',3A6,10X,'PAGE',
     1I3,//,45X,'STUDY-ID    NCASE    NTASK    NDUTY',/,45X,A6,A2,5X,I4,
     A 5X, I4, 6X, I3, //, 13X, 'REPORT TITLE',
     2 38X,'REPORT-ID   SOURCE FILE     MEMBERS    REPORT'
     3,/,102X,'NUMBER',/)
  101 FORMAT(13X,8A6,5X,A6,3X,'HISTORY',10X,I6,7X,I3)
  102 FORMAT(13X,8A6,5X,A6,3X,'KPATH - ',A6,3X,I6,7X,I3)
  103 FORMAT('0',12X,'READ ERROR OCCURRED ON NEXT JOB DESCRIPTION')
  104 FORMAT('0',12X,'END OF JOB DESCRIPTION FILE',/,'1')
  105 FORMAT('1',12X,'NO JOB DESCRIPTIONS FOUND IN FILE')
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      PARAMETER MAXCAS = 7000
      PARAMETER MAXTKD = 1726
      COMMON LINKPH(MAXCAS),LCOMP(MAXTKD),LMIST(MAXTKD),LTASK(MAXTKD),
     1 KDUTY(26), NMASK(200), ICARD(14), NHEAD, LABEL(42), KSWT,
     2  JSEQ, NCOM, ICOM1(50), ISKIP(MAXTKD)
      DIMENSION LOCTIT(MAXTKD)
      EQUIVALENCE (LMIST,LOCTIT)
C  NOTE THAT LMIST AND LTASK HAVE LARGE DIMENSIONS BECAUSE THE
C  CORE SPACE IS USED FOR DIFFERENT PURPOSES IN THE SUBROUTINES
      CALL SETUP9(10,25,26,0,0,0,20,0,0,0,0,JSEQ)
      INP = 10
      IJD = 25
      ISC = 20
      IRP = 26
      READ(INP) NCOM, (ICOM1(I),I=1,NCOM)
      NCASE = ICOM1(3)
      NTASK = ICOM1(12)
      NDUTY = ICOM1(9)
      NTD = NTASK + NDUTY
      NDICT = ICOM1(4)
      CALL RESET
      READ(5,800) ID1, ID2, KHEAD
800   FORMAT(7X,A6,A2,1X,A1)
      CALL PRINTC
      IF(ICOM1(1).EQ.ID1.AND.ICOM1(2).EQ.ID2) GO TO 1
      WRITE(6,801) ICOM1(1), ICOM1(2), ID1, ID2
801   FORMAT(' STUDY-ID ON TAPE = ',A6,A2,'.  ON CONTROL CARD = ',
     1 A6,A2,'.  RUN ABORTS.')
      CALL ERTRAN(2)
1     CONTINUE
      CALL BCDBIN(KHEAD,NHEAD,1,$2)
      IF(NHEAD.LE.3) GO TO 3
2     CALL ERRMSG(7,'*** INVALID NUMBER OF HEADING CARDS ABOVE.')
      CALL ERTRAN(2)
3     CONTINUE
      IF(NHEAD.EQ.0) GO TO 11
      DO 10 LCARD = 1, NHEAD
      IBEGIN = (LCARD-1)*14+1
      ISTOP = IBEGIN + 13
      READ(5,802) (LABEL(I),I=IBEGIN,ISTOP)
802       FORMAT(13A6,A2)
      CALL PRINTC
10    CONTINUE
11    CONTINUE
      DEFINE FILE 29(NCASE,NTD,U,NEXT)
      ITK = 29
      DEFINE FILE  9(NTD,99,U,NEXT1)
      ITL = 9
      READ(INP) N, (ISKIP(I),I=1,N), NDUT, (KDUTY(I),I=1,NDUT)
      DO 4 I = 1, NDICT
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      READ(INP)
    4 CONTINUE
      DO 5 ICASE = 1, NCASE
      READ(INP)  K, (LCOMP(I),I=1,K),  L, (LHIST(I),I=1,L),
     1 M, (LTASK(I),I=1,M)
      WRITE(29'ICASE) M, (LTASK(I),I=1,M)
C  ESTABLISH LINKAGE IN KPATH ORDER.
C   LEFT-HALF = INTERNAL KPATH SEQUENCE NUMBER
C   RIGHT-HALF= CASE-ID ASSIGNED BY INPSTD
      LINKPH(ICASE) = LCOMP(1)
      FLD(0,18,LINKPH(ICASE)) = LCOMP(3)
    5 CONTINUE
C  READ FIRST END-OF-DATA RECORD
      READ(INP)
C
C  BEGIN READING TASK TITLE BLOCKS
      IBLOCK = 0
    6 CONTINUE
      READ(INP) K, (LTASK(I),I=1,K)
      IF(LTASK(1).EQ.'END OF'.AND.LTASK(2).EQ.' DATA') GO TO 7
      IBLOCK = IBLOCK + 1
      WRITE(ISC) K, (LTASK(I),I=1,K)
      GO TO 6
    7 CONTINUE
      REWIND ISC
      BACKSPACE INP
      BACKSPACE INP
      BACKSPACE INP
      READ(INP) N, (LOCTIT(ITASK),ITASK=1,N)
      IREC = 0
      DO 9 ITITL = 1, NTD
      JREC = FLD(0,18,LOCTIT(ITITL))
      IF(IREC.EQ.JREC) GO TO 8
      READ(ISC) N, (LTASK(I),I=1,N)
      IREC = JREC
    8 CONTINUE
      LOCCON = FLD(18,18,LOCTIT(ITITL))
      LASWRD = FLD(0,18,LTASK(LOCCON))
      LENGTH = LASWRD - LOCCON
      ISTART = LOCCON + 1
      WRITE(ITL'ITITL) LENGTH, (LTASK(I),I=ISTART,LASWRD)
    9 CONTINUE
      CALL FREE(INP)
      CALL FREE(ISC)
      CALL JOBGP1
      STOP
      SUBROUTINE JOBGP1
      COMMON LINKPH(MAXCAS), TASK(MAXTKD), AVE(MAXTKD), ANUMP(MAXTKD),
     1 KDUTY(26), NMASK(200), ICARD(14), NHEAD, LABEL(42), KSWT,
     2  JSEQ, NCOM, ICOM1(50), INDTSK(MAXTKD)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      EQUIVALENCE (ICOM1(16),LOCASE)
      EQUIVALENCE (ICOM1(17),MBRS)
      EQUIVALENCE (ICOM1(18),IDGRP)
      EQUIVALENCE (ICOM1(24),IDSP)
      EQUIVALENCE (ICOM1(27),ITAPS)
      EQUIVALENCE (ICOM1(28),PMAX)
      EQUIVALENCE (ICOM1(29),MAXT)
      EQUIVALENCE (ICOM1(34),KSW)
      NCASE = ICOM1(3)
      NTASK = ICOM1(12)
      NDUTY = ICOM1(9)
      NTD = NTASK + NDUTY
      ITL = 9
      ITK = 29
 1    CONTINUE
      CALL ZERMSK(NCASE,NMASK,LEN)
 11   READ(5,800,END=99) LOW,KHI, LSW,LTAPS, MIN,MAX,MAXT, PMAX, LSWT
 800  FORMAT(A4,1X,A4,2(1X,A1),3(A5),F4.0,1X,A1)
      GO TO 2
 900  CONTINUE
      CALL RESET
      CALL PRINTC
      CALL ERRMSG(8,'*** INVALID NUMERIC ABOVE, REQUEST BYPASSED.')
      GO TO 11
 2    CALL BCDBIN(LOW,LOW,4,$900)
      CALL BCDBIN(KHI,KHI,4,$900)
      CALL BCDBIN(MIN,MIN,5,$900)
      CALL BCDBIN(MAX,MAX,5,$900)
      CALL BCDBIN(MAXT,MAXT,5,$900)
      CALL BCDBIN(LSW,LSW,1,$900)
      CALL BCDBIN(LTAPS,LTAPS,1,$900)
      CALL BCDBIN(LSWT,LSWT,1,$900)
          IF(KHI.EQ.0) KHI = LOW
      KSW = LSW
      IF(LSW.LE.0.OR.LSW.GT.3) KSW = 3
      ITAPS = LTAPS
      IF(LTAPS.GT.1) ITAPS = 0
      ITAPS = 2 - ITAPS
      IF(MIN.EQ.0) MIN = 5
      IF(MAX.EQ.0) MAX = NCASE
      IF(MAXT.EQ.0) MAXT = NTASK
      IF(PMAX.LE.0.0) PMAX = 100.0
      KSWT = LSWT
      IF(LSWT.NE.1) KSWT = 0
      KSWT = KSWT + 1
      KPSEQ = LOW
 20   INDEX = 1
 21   IDGRP = NCASE
      MBRS = 1
      LOCASE = INDEX
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
22    INDEX = INDEX + 1
      IF(INDEX.GT.NCASE) GO TO 24
      ISTAGE = FLD(0,18,LINKPH(INDEX)) - 1
      IF(ISTAGE.LT.KPSEQ) GO TO 24
      IF(ISTAGE.GT.IDGRP) GO TO 23
      IDGRP = ISTAGE
23    MBRS = MBRS + 1
      GO TO 22
24    IF(IDGRP.EQ.KPSEQ) GO TO 27
25    IF(INDEX.LE.NCASE) GO TO 21
26    KPSEQ = KPSEQ + 1
      IF(KPSEQ.LE.KHI) GO TO 20
      GO TO 11
27    IF(MBRS.GE.MIN.AND.MBRS.LE.MAX) GO TO 28
      WRITE(6,801) IDGRP, MBRS, MIN, MAX
801   FORMAT(' GROUP ',I6,' HAD ',I6,' MEMBERS.  MIN = ',I6,
     1 ' MAX = ',I6,'.  REQUEST BYPASSED.')
      GO TO 26
28    CONTINUE
      LASCAS = LOCASE + MBRS - 1
      CALL ZERO(AVE(1),NTD)
      CALL ZERO(ANUMP(1),NTD)
      CALL ZEROI(NMASK(1),200)
      DO 39 ICASE = LOCASE, LASCAS
      READ(ITK'ICASE) NRESP, (INDTSK(I),I=1,NRESP)
      CALL ZERO(TASK(1),MAXTKD)
      DO 30 IRESP = 1, NRESP
      ITASK = FLD(0,18,INDTSK(IRESP))
      TASK(ITASK) = FLD(18,18,INDTSK(IRESP))/1000.0
30    CONTINUE
      DO 32 IDUTY = 1, NDUTY
      LOWTSK = FLD( 0,18,KDUTY(IDUTY))
      IHITSK = FLD(18,18,KDUTY(IDUTY))
      DO 31 ITASK = LOWTSK, IHITSK
      LOCDTY = NTASK + IDUTY
      TASK(LOCDTY) = TASK(LOCDTY) + TASK(ITASK)
31    CONTINUE
32    CONTINUE
      CALL SETMSK(ICASE,NMASK)
      DO 33 ITASK = 1, NTD
      IF(TASK(ITASK).LE.0.0) GO TO 33
      ANUMP(ITASK) = ANUMP(ITASK) + 1.0
      AVE(ITASK) = AVE(ITASK) + TASK(ITASK)
33    CONTINUE
C
C END OF DATA COMPILATION LOOP
C
39    CONTINUE
      CALL JOBPRT
          IF(KPSEQ.LT.KHI) GO TO 26
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIF FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      GO TO 1
99    CONTINUE
      CALL RESET
      CALL ERRMSG(4,'*** END OF JOBGRP RUN ')
      RETURN
      SUBROUTINE JOBPRT
      COMMON LINKPH(MAXCAS), TASK(MAXTKD), AVE(MAXTKD), ANUMP(MAXTKD),
     1 KDUTY(26), NMASK(200), ICARD(14), NHEAD, LABEL(42), KSWT,
     2  JSEQ, NCOM, ICOMI(50), INDTSK(MAXTKD)
      DIMENSION LABL(6,2)
      DIMENSION ITYP(2)
      DIMENSION ITITLE(99), LTITLE(11,9)
      EQUIVALENCE (INDTSK,ITITLE), (INDTSK,LTITLE)
      DATA LABL/6*' ','TASK  SEQ   NO    .      .      .      '/
      DATA ITYP/' DUTY', ' TASK'/
      EQUIVALENCE (ICOMI(16),LOCASE)
      EQUIVALENCE (ICOMI(17),MBRS)
      EQUIVALENCE (ICOMI(18),IDGRP)
      EQUIVALENCE (ICOMI(24),IDSP)
      EQUIVALENCE (ICOMI(27),ITAPS)
      EQUIVALENCE (ICOMI(28),PMAX)
      EQUIVALENCE (ICOMI(29),MAXT)
      EQUIVALENCE (ICOMI(34),KSW)
          NCASE = ICOMI(3)
          NTASK = ICOMI(12)
          NDUTY = ICOMI(9)
      NTD = NTASK + NDUTY
      LASCAS = LOCASE + MBRS - 1
C   GENERATE REPORT TITLE
      ENCODE(800,ICARD) IDGRP, LOCASE, LASCAS
800   FORMAT('STAGE = ',I4,'   KPATH ORDER FROM ',I4,' TO ',I4)
C   GENERATE REPORT ID FROM ID OF GROUP
      ENCODE(801,IDSP) IDGRP
801   FORMAT('GP',J4)
      AMBRS = MBRS
      DO 5 ITASK = 1, NTD
      TASK(ITASK) = AVE(ITASK)/AMBRS
5     CONTINUE
      IF(ITAPS.NE.2) GO TO 6
      ICOMI(35) = 'JOBGRP'
      ICOMI(36) = 1
      ENCODE(802,ICOMI(37)) (ICARD(I),I=1,8)
802   FORMAT(8A6)
      LENMSK = (NCASE+34)/35
      CALL ERTRAN(9,IDATE,ITIME)
      ICOMI(46) = IDATE
      ICOMI(47) = ITIME
      ICOMI(50) = 'FIXBJD'
      WRITE(IJD) IDSP,NTD,(TASK(I),I=1,NTD),LENMSK,(NMASK(I),I=1,LENMSK)
     1, JSEQ, NCOM, (ICOMI(I),I=1,NCOM), (AVE(I),I=1,NTD),
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
        2  (ANUMP(I),I=1,NTD)
  6     IDUT = 1
        ITSK = 2
        IJD = 25
        IRP = 26
        ITL = 9
        ITK = 29
        IF(KSW.EQ.2) IDUT = 2
        IF(KSW.EQ.1) ITSK = 1
        DO 50 IPRINT = IDUT, ITSK
        ISEQ = 0
        IFIRST = 1
        LAST = NTASK
        NOPRT = 1
        GO TO (7,8), IPRINT
  7     IFIRST = NTASK + 1
        LAST = NTD
  8     CONTINUE
        LPAGE = 1
        DO 13 IWRITE = 1, ITAPS
        IUNIT = 6
        IF(IWRITE.EQ.2) IUNIT = IRP
        WRITE(IUNIT,803) IDGRP, LOCASE, LASCAS, IDSP, LPAGE
  803   FORMAT('ISTAGE = ',I4,'   KPATH ORDER FROM ',I4,' TO ',
       1 I4,33X,A6,1X,'PAGE ',I4,/)
        IF(NHEAD.EQ.0) GO TO 10
        DO 9 IHEAD = 1, NHEAD
        ISTART = (IHEAD-1)*14 + 1
        ISTOP  = ISTART + 13
        WRITE(IUNIT,804) (LABEL(I),I=ISTART,ISTOP)
  804   FORMAT(1X,13A6,A2)
  9     CONTINUE
        WRITE(IUNIT,805)
  805   FORMAT(1X)
  10    CONTINUE
        WRITE(IUNIT,806) ITYP(IPRINT), NCASE, NTASK, NDUTY, MBRS
  806   FORMAT(1X,A6,'JOB DESCRIPTION',6X,'CASES    TASKS',
       1 '   DUTIES     MBRS',/,24X,4I9,/)
        WRITE(IUNIT,807) (LABL(I,IPRINT),I=1,6)
  807   FORMAT(/27X, 66HCUMULATIVE SUM OF AVERAGE PERCENT TIME SPENT BY A
       1LL MEMBERS.......,5X,A4,
       2         /27X, 59HAVERAGE PERCENT TIME SPENT BY ALL MEMBERS........
       3..........,6X,1H.,5X,A3,
       4         /27X, 52HAVERAGE PERCENT TIME SPENT BY MEMBERS PERFORMING.
       5...,6X,1H.,6X,1H.,5X,A2,
       6         /27X, 45HPERCENT OF MEMBERS PERFORMING.................,6X,
       71H.,6X,1H.,6X,1H.,6X,A1,/65X,4(6X,1H.),6X,A1,/6H D-TSK,
       821X,15HDUTY/TASK TITLE,23X,4(6X,1H.),6X,A1,/)
  13    CONTINUE
        LINE = 12 + NHEAD
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
14    CONTINUE
      ATIT4 = 0.0
      DO 49 ITASK = IFIRST, LAST
      AMAX = 0.0
      DO 17 LOOK = IFIRST, LAST
      GO TO (16,15), KSWT
15    IF(ANUMP(LOOK).LE.AMAX) GO TO 17
         AMAX = ANUMP(LOOK)
         KEEP = LOOK
         GO TO 17
16    IF(AVE(LOOK).LE.AMAX) GO TO 17
         AMAX = AVE(LOOK)
         KEEP = LOOK
17    CONTINUE
      IF(AMAX.LE.0.0) GO TO 50
        ATIT1 = ANUMP(KEEP)/AMBRS * 100.0
        ATIT2 = AVE(KEEP)/ANUMP(KEEP)
        ATIT3 = TASK(KEEP)
        ATIT4 = ATIT3 + ATIT4
      READ(ITL'KEEP) LEN, (ITITLE(I),I=1,LEN)
      LINES = (LEN+10)/11
      LINE = LINE + LINES
      IF(LINE.LT.55) GO TO 20
      LINE = 3
      LPAGE = LPAGE + 1
      DO 18 IWRITE = 1, ITAPS
      IUNIT = 6
      IF(IWRITE.EQ.2) IUNIT = IRP
      IF(IUNIT.EQ.6.AND.NOPRT.EQ.2) GO TO 18
      WRITE(IUNIT,808) IDSP, LPAGE
808   FORMAT('1',77X,A6,' PAGE ',I4,/)
18    CONTINUE
20    CONTINUE
      ISEQ = ISEQ + 1
      DO 24 IWRITE = 1, ITAPS
      IUNIT = 6
      IF(IWRITE.EQ.2) IUNIT = IRP
C   TEST FOR EVERY FIFTH LINE FOR TASK NUMBER
      ICHK = MOD(ISEQ,5)
      IF(ATIT4.LE.PMAX.AND.ISEQ.LE.MAXT) GO TO 21
      IF(IPRINT.EQ.1) GO TO 22
      NOPRT = 2
      GO TO (24,21), IWRITE
21    IF(ATIT1.LE.0.0.AND.ATIT2.LE.0.0.AND.ATIT3.LE.0.0) GO TO 24
      IF(IPRINT.EQ.1) GO TO 22
      IF(ICHK.NE.0) GO TO 22
      WRITE(IUNIT,809) (LTITLE(I,1),I=1,11),ATIT1,ATIT2,ATIT3,ATIT4,ISEQ
809   FORMAT(1X,11A6,4F7.2,1X,I4)
      GO TO 23
22    WRITE(IUNIT,810) (LTITLE(I,1),I=1,11),ATIT1,ATIT2,ATIT3,ATIT4
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

JOBGRP JOB DESCRIPTIONS OF GROUPS AT GIVEN STAGES

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
810   FORMAT(1X,11A6,4F7.2)
23    IF(LINES.EQ.1) GO TO 24
      DO 25 ILINE = 2, LINES
      WRITE(IUNIT,810) (LTITLE(I,ILINE),I=1,11)
25    CONTINUE
24    CONTINUE
      ANUMP(KEEP) = -ANUMP(KEEP)
      AVE(KEEP) = -AVE(KEEP)
49    CONTINUE
      DO 48 ITASK = IFIRST, LAST
      ANUMP(ITASK) = ABS(ANUMP(ITASK))
      AVE(ITASK) = ABS(AVE(ITASK))
48    CONTINUE
50    CONTINUE
      IF(ITAPS.EQ.1) RETURN
      CALL REPEND(ICOM1(1),IDSP,ICARD,JSEQ)
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
      IMPLICIT INTEGER(O)
      DIMENSION KOMM(50),KARD(18),ITL(7),NIGH(14,3),IDV(30),
     1 NDC(30),ITLV(3,30),KARV(4,99,30),ITLR(8),LTHV(1000),
     2 LTDU(26),LDICT(15,16),IDIC(2),IMAT(2),MORF(2,2000),
     3 MAFM(2),MIFM(2),IVFM(2),LREL(30),LENV(30),LFOR(2,30),
     4 JCV(500),JMD(1000),JTD(1700),LPRINT(3,30),NIRP(22),
     5 IBUF(700),JBUF(7000),IFOR(4),KASE(7000),KVIS(7000),
     6 ISORT(4),LOAR(7000),KTIT(465),LOCTSK(1726),LTIT(11,9),
     7 LTIA(99)
      DIMENSION IUSE24(3),IJOB(200),ISP(200),ITX(8),IALLV(13)
      EQUIVALENCE (IALLV,IDSORT),(IALLV(2),IDSPL),(IALLV(3),IRSX),
     1 (IALLV(4),IPS),(IALLV(5),ISEED),(IALLV(6),ITX)
      EQUIVALENCE (IDR,KARD(11)),(ITL(1),KARD(12)),(KVIS,LOAR)
      EQUIVALENCE (LTIT(1,1),LTIA(1))
      DATA IUSE24/'@USE 28.,24. . '/
      IFOR(1) = '('
      IFOR(4) = ')'
      CALL SETUP(10,27,26,15,0,0,24,0,0,0,0,JSEQ)
      ENCODE(8560,ITL)
 8560 FORMAT(42X)
      LIO = 10
      LOO = 26
      LDO = 9
      LD1 = 29
      LD2 = 8
      LD3 = 24
      ISU = 15
      IDSORT = 0
      REWIND LIO
      IPASS = 1
C  READ I.D. CARD.
      READ(5,8020,END=7000) NAME,IDS1,IDS2
 8020 FORMAT(A6,1X,A6,A2)
      DECODE(8030,IDS1) IS4
 8030 FORMAT(A4)
C  READ COMM. AND VERIFY STUDY I.D. .
      READ(LIO) N,(KOMM(INDEX),INDEX = 1,N)
      ID1 = KOMM(1)
      ID2 = KOMM(2)
      NCASE = KOMM(3)
      NDICT = KOMM(4)
      MID = KOMM(6)
      NTASK = KOMM(12)
      NSKIP = NDICT + 2
      IF(IDS1.EQ.ID1 .AND. IDS2.EQ.ID2) GO TO 50
      WRITE(6,8040) IDS1,IDS2,ID1,ID2,NAME
 8040 FORMAT('1   STUDY I.D. NO MATCH BETWEEN CONTROL CARD (',
     1 A6,A2,') AND COMMUNICATION REGION (',A6,A2,').'/4X,A6,
     2 ' ABORTS.')
      CALL ERTRAN(2)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
50    CONTINUE
      WRITE(ISU) N,(KOMM(I),I=1,N)
C  DEFINE WORK FILES.
      JERK = 150 + NTASK + 1 + 1
      DEFINE FILE LD1(NCASE,JERK,U,NEXT)
      DEFINE FILE LD2(NTASK,99,U,NERF)
      DEFINE FILE LD0(2000,4,U,NULL)
C
C  PROCESSING LOOP
C
100   CONTINUE
      IFLAGA = 0
C  READ SET CARD.
      READ(5,8060,END=7000) (KARD(INDEX),INDEX = 1,18)
8060  FORMAT(A3,1X,2(A1,1X),A4,1X,3(A1,1X),2(A4,1X),A1,1X,A6,1X,7A6)
      IF(KARD(1).NE.'SET') GO TO 7100
      CALL BCDBIN(KARD(2),NHCS,1,$7100)
      IF(NHCS.GT.3) GO TO 7100
      IF(NHCS.NE.0) GO TO 120
C  NO HEADING CARDS
      ASSIGN 1800 TO JUMPA
      ASSIGN 6120 TO JUMPG
      ASSIGN 6290 TO JUMPN
      GO TO 140
120   CONTINUE
C  1 - 3 HEADING CARDS
      ASSIGN 1700 TO JUMPA
      ASSIGN 6110 TO JUMPG
      ASSIGN 6280 TO JUMPN
140   CONTINUE
C  PICK REPORT TERMINATION OPTION.
      CALL BCDBIN(KARD(3),OPTN,1,$400)
      IF(OPTN.EQ.0 .OR. OPTN.GT.3) GO TO 400
      CALL BCDBIN(KARD(4),KUTOFF,4,$150)
      IFLAGA = 1
150   CONTINUE
      IFLAGA = IFLAGA + 1
C  GO TO (ERROR/DEFAULT,SPECIFIED) OPTION.
      IF(OPTN.EQ.2) GO TO (400,450),IFLAGA
      IF(OPTN.EQ.3) GO TO (300,350),IFLAGA
      GO TO (200,250),IFLAGA
C  TASK CUTOFF
200   CONTINUE
      KUTOFF = NTASK
250   CONTINUE
      IF(KUTOFF.GT.NTASK) KUTOFF = NTASK
      ASSIGN 6160 TO JUMPI
      ASSIGN 6330 TO JUMPP
      ASSIGN 6250 TO JUMPK
      ASSIGN 6420 TO JUMPR
```

228

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      ASSIGN 6184 TO JUMPL
      ASSIGN 6354 TO JUMPS
      GO TO 600
C  CUMULATIVE PER CENT CUTOFF
300   CONTINUE
      KUTOFF = 100
350   CONTINUE
      IF(KUTOFF.GT.100) KUTOFF = 100
      KUTOFF = KUTOFF * 1000
      ASSIGN 6170 TO JUMPI
      ASSIGN 6340 TO JUMPP
      ASSIGN 6210 TO JUMPK
      ASSIGN 6380 TO JUMPR
      ASSIGN 6184 TO JUMPL
      ASSIGN 6354 TO JUMPS
      GO TO 600
C  FULL REPORT
400   CONTINUE
      ASSIGN 6170 TO JUMPI
      ASSIGN 6340 TO JUMPP
      ASSIGN 6250 TO JUMPK
      ASSIGN 6420 TO JUMPR
      ASSIGN 6184 TO JUMPL
      ASSIGN 6354 TO JUMPS
      GO TO 600
C  PAGE CUTOFF
450   CONTINUE
      ASSIGN 6170 TO JUMPI
      ASSIGN 6340 TO JUMPP
      ASSIGN 6250 TO JUMPK
      ASSIGN 6420 TO JUMPR
      ASSIGN 6182 TO JUMPL
      ASSIGN 6352 TO JUMPS
600   CONTINUE
C  'D-TSK'/SEQUENTIAL TASK OPTION
      CALL BCDBIN(KARD(5),OPTI,1,$700)
      IF(OPTI.NE.1) GO TO 700
      LHDT = '  TSK'
      ASSIGN 6190 TO JUMPJ
      ASSIGN 6360 TO JUMPQ
      GO TO 800
700   CONTINUE
      LHDT = 'D-TSK'
      ASSIGN 6200 TO JUMPJ
      ASSIGN 6370 TO JUMPQ
800   CONTINUE
C  NUMBER OF COPIES OF PRINTED REPORT
      CALL BCDBIN(KARD(6),NCOP,1,$900)
      GO TO 1000
900   CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
       NCOP = 1
 1000  CONTINUE
       IF(NCOP.EQ.0) NCOP = 1
       CALL BCDBIN(KARD(7),OPST,1,$1100)
       IF(OPST.EQ.2) GO TO 1300
       IF(OPST.EQ.1) GO TO 1200
C  PRINT IN INPUT SEQUENCE.
 1100  CONTINUE
       ASSIGN 3080 TO JUMPB
       ASSIGN 3900 TO JUMPE
       ASSIGN 6000 TO JUMPF
       ASSIGN 6020 TO JUMPFI
       ASSIGN 6500 TO JUMPT
       GO TO 1400
C  PRINT IN RANDOM SEQUENCE.
 1200  CONTINUE
       ASSIGN 3080 TO JUMPB
       ASSIGN 3900 TO JUMPE
       ASSIGN 5050 TO JUMPF
       ASSIGN 6020 TO JUMPFI
       ASSIGN 6500 TO JUMPT
       GO TO 1400
C  PRINT IN ASCENDING SEQUENCE.
 1300  CONTINUE
       IF(KARD(9).EQ.' ') GO TO 1350
       ASSIGN 3000 TO JUMPB
       ASSIGN 4000 TO JUMPE
       ASSIGN 5100 TO JUMPF
       ASSIGN 5150 TO JUMPFA
       ASSIGN 6040 TO JUMPFI
       ASSIGN 6530 TO JUMPT
       GO TO 1400
 1350  CONTINUE
C  NO MINOR SORT FIELD
       ASSIGN 3040 TO JUMPB
       ASSIGN 4100 TO JUMPE
       ASSIGN 5100 TO JUMPF
       ASSIGN 5350 TO JUMPFA
       ASSIGN 6040 TO JUMPFI
       ASSIGN 6530 TO JUMPT
 1400  CONTINUE
       IF(KARD(10).NE.'1') GO TO 1500
C  NO REPORT FILE OUTPUT
       ASSIGN 6600 TO JUMPM
       GO TO 1600
 1500  CONTINUE
C  ADD REPORT FILE ENTRIES.
       ASSIGN 6270 TO JUMPM
 1600  CONTINUE
       GO TO JUMPA,(1700,1800)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
1700  CONTINUE
C  READ HEADING CARDS.
      READ(5,8080,END=7000) ((NIGH(INDEX,INDEX1),INDEX = 1,14),
     1 INDEX1 = 1,NHCS)
8080  FORMAT(13A6,A2)
1800  CONTINUE
C  READ NEXT CARD.
      READ(5,8100,END=7000) IDC,NSPEC,IDV1,JSSP,IFSP,IRSO,IPS,IX,IY,LAST
8100  FORMAT(A3,1X,A2,1X,A4,1X,A1,1X,A4,1X,A1,1X,A3,1X,A1,1X,A1,48X,A4)
      IF(IDC.NE.'VAR') GO TO 2200
C  "VAR" CARD
      ITRACE = 0
      IF(IY.NE.' ') ITRACE = 1
      IF(ITRACE.EQ.1) IDSORT = IDSORT + 1
      IRSX = 0
      IF(IRSO.EQ.'1') IRSX = 1
      IF(IRSO.EQ.'2') IRSX = 2
      IF(IRSX.EQ.0) GO TO 1801
      CALL BCDBIN(IPS,IPS,3,$7100)
      RPS = IPS
      RPS = RPS/100.0
      IRSN = IPS
1801  CALL BCDBIN(NSPEC,NSPEC,2,$7100)
      CALL BCDBIN(IFSP,IBSSPN,4,$7100)
      NLSSP = 0
      IF(JSSP.NE.' ') NLSSP = 4
      IF(NSPEC.EQ.0) GO TO 2450
      IF(NSPEC.GT.30) GO TO 7100
      ASSIGN 3090 TO JUMPB1
      ASSIGN 3190 TO JUMPC
      ASSIGN 3500 TO JUMPD
      ASSIGN 6130 TO JUMPH
      ASSIGN 6300 TO JUMPO
C  READ "SPEC" CARDS.
      DO 2100 INDEXA = 1,NSPEC
      READ(5,8120,END=7000) IDV(INDEXA),NDC(INDEXA),
     1 (ITLV(INDEX,INDEXA),INDEX = 1,3)
8120  FORMAT(A4,1X,A2,1X,3A6)
      CALL BCDBIN(NDC(INDEXA),NDC(INDEXA),2,$7100)
      NDCS = NDC(INDEXA)
      IF(NDCS.EQ.0) GO TO 2000
C  READ "DECODE" CARDS.
      DO 1900 INDEXB = 1,NDCS
      READ(5,8140,END=7000) (KARV(INDEX,INDEXB,INDEXA),INDEX = 1,4)
8140  FORMAT(A6,1X,3A6)
1900  CONTINUE
2000  CONTINUE
2100  CONTINUE
      GO TO 2500
C  NON-"VAR" CARD
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 2200  CONTINUE
       CALL BCDBIN(LAST,LAST,4,$2300)
       IF(LAST.NE.0) GO TO 2400
 2300  CONTINUE
 C  LOOP OR CARD ERROR.
       IF(IDC.NE.'END') GO TO 7100
       GO TO 100
 2400  CONTINUE
 C  REREAD "REQUEST" CARD.
       READ(0,8160,END=7000) IDUST,IDRP,ITLR,JSEQR
 8160  FORMAT(A4,9A6,18X,A4)
 C  NO "VAR" CARD
       NSPEC = 1
       ASSIGN 3400 TO JUMPBI
       ASSIGN 3870 TO JUMPD
       ASSIGN 6150 TO JUMPH
       ASSIGN 6320 TO JUMPO
       GO TO 2600
 2450  CONTINUE
 C  NO "SPEC" CARDS
       NSPEC = 1
       ASSIGN 3090 TO JUMPBI
       ASSIGN 3400 TO JUMPC
       ASSIGN 3870 TO JUMPD
       ASSIGN 6150 TO JUMPH
       ASSIGN 6320 TO JUMPO
 2500  CONTINUE
 C  READ "REQUEST" CARD.
       READ(5,8160,END=7000) IDUST,IDRP,ITLR,JSEQR
 2600  CONTINUE
       CALL BCDBIN(JSEQR,JSEQR,4,$7100)
       IF(JSEQR.EQ.0) GO TO 7100
 C  FIND "END" CARD.
       READ(5,8180,END=7000) LWD
 8180  FORMAT(A3)
       IF(LWD.EQ.'END') GO TO 2800
       WRITE(6,8200) LWD
 8200  FORMAT('1    "',A3,'" CARD IGNORED.  SEARCHING FOR "END" CARD')
 2700  CONTINUE
       READ(5,8180,END=7000) LWD
       IF(LWD.NE.'END') GO TO 2700
 2800  CONTINUE
 C  GET CASE MASK.
       CALL GETMSM(JSEQR,ID1,ID2,MID,$100,MEMBX,IDSPL,ITX)
       IF(ITLR(1).EQ.' ') ENCODE(8080,ITLR) ITX
       IF(IRSX.EQ.0) GO TO 2801
       IF(IRSX.EQ.1) IRSN = MEMBX*RPS + .5
       IF(IRSN.LT.1) IRSN = 1
       CALL ERTRAN(9,IDATE,ITIME)
       ISEED =  IABS(IDATE + ITIME + 31)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      CALL IRAND(ISEED,MEMBX,IRSN)
      WRITE(6,8201) IRSN,MEMBX,IDSPL,ISEED
 8201 FORMAT('1JOBIND WILL RANDOMLY SELECT ',I5,' OF ',I5,' MEMBERS',
     1 ' IN ',A6,' USING ',O12,' (OCTAL) AS A SEED.')
 2801 IF(IPASS.NE.1) GO TO 2900
C  ON FIRST PASS OF LOOP, GET CONTROL INFO. AND DICTIONARY FORMATS.
      IPASS =  2
C  READ RELATIVE LOCATION TABLES.
      READ(LIO) NHV,(LTHV(INDEX),INDEX = 1,NHV),NDUTY,(LTDU(INDEX),
     1 INDEX = 1,NDUTY)
      LOREL = 0
C  BUILD VARIABLE FORMAT TABLE INDEXED BY VID
C  (BACKGROUND VID'S BIASED BY LOREL)
      DO 2890 INDEXA = 1,NDICT
      READ(LIO) NINREC,((LDICT(INDEX,INDEX1),INDEX = 1,15),
     1 INDEX1 = 1,NINREC)
      DO 2880 INDEXB = 1,NINREC
      DECODE(8210,LDICT(1,INDEXB)) LETR,INDEXC
 8210 FORMAT(A1,I3)
      IDIC(1) = LDICT(14,INDEXB)
      IDIC(2) = LDICT(15,INDEXB)
      CALL FORMAT(IDIC,IMAT,NCHR,ITYP,I1,I2,$7600)
      IF(LETR.EQ.'C') GO TO 2860
      INDEXC = INDEXC + LOREL
      GO TO 2870
 2860 CONTINUE
      LOREL = LOREL + 1
 2870 CONTINUE
      MORF(1,INDEXC) = IMAT(1)
      MORF(2,INDEXC) = IMAT(2)
 2880 CONTINUE
 2890 CONTINUE
      GO TO 2970
C  ON NONFIRST PASS OF LOOP, ALIGN HISTORY/KPATH FILE TO CASE DATA.
 2900 CONTINUE
      REWIND LIO
      DO 2950 INDEXA = 1,NSKIP
      READ(LIO) MASH
 2950 CONTINUE
 2970 CONTINUE
C  GET RELATIVE LOCATIONS, LENGTHS, AND FORMATS FOR SORT AND
C   VISUAL I.D. VARIABLES.
      GO TO JUMPB,(3000,3040,3080,3400)
 3000 CONTINUE
      DECODE(8212,KARD(9)) LETR,NVID
      CALL BCDBIN(NVID,NVID,3,$7200)
      IF(LETR.NE.'V') GO TO 3020
      MIRL = -FLD(18,18,LTHV(NVID))
      MILE =  FLD( 0,18,LTHV(NVID))
      NVID = NVID + LOREL
```

LINE # TEXT...

```
      MIFM(1) = MORF(1,NVID)
      MIFM(2) = MORF(2,NVID)
      GO TO 3040
3020  CONTINUE
      MIRL = NVID
      MILE = 1
      MIFM(1) = MORF(1,NVID)
      MIFM(2) = MORF(2,NVID)
3040  CONTINUE
      DECODE(8212,KARD(8)) LETR,NVID
      CALL BCDBIN(NVID,NVID,3,$7200)
      IF(LETR.NE.'V') GO TO 3060
      MARL = -FLD(18,18,LTHV(NVID))
      MALE =  FLD( 0,18,LTHV(NVID))
      NVID = NVID + LOREL
      MAFM(1) = MORF(1,NVID)
      MAFM(2) = MORF(2,NVID)
      GO TO 3030
3060  CONTINUE
      MARL = NVID
      MALE = 1
      MAFM(1) = MORF(1,NVID)
      MAFM(2) = MORF(2,NVID)
3080  CONTINUE
      GO TO JUMPB1,(3090,3400)
3090  CONTINUE
      DECODE(8212,IDV1) LETR,NVID
      CALL BCDBIN(NVID,NVID,3,$7200)
      IVRL = -FLD(18,18,LTHV(NVID))
      IVLE =  FLD( 0,18,LTHV(NVID))
      NVID = NVID + LOREL
      IVFM(1) = MORF(1,NVID)
      IVFM(2) = MORF(2,NVID)
3100  CONTINUE
      GO TO JUMPC,(3190,3400)
C  GET RELATIVE LOCATIONS, LENGTHS, AND FORMATS FOR "SPEC" VARIABLES.
3190  CONTINUE
      DO 3300 INDEXA = 1,NSPEC
      DECODE(8212,IDV(INDEXA)) LETR,NVID
8212  FORMAT(A1,A3)
      CALL BCDBIN(NVID,NVID,3,$7200)
      IF(LETR.NE.'V') GO TO 3200
      LREL(INDEXA) = -FLD(18,18,LTHV(NVID))
      LENV(INDEXA) =  FLD( 0,18,LTHV(NVID))
      NVID = NVID + LOREL
      LFOR(1,INDEXA) = MORF(1,NVID)
      LFOR(2,INDEXA) = MORF(2,NVID)
      GO TO 3300
3200  CONTINUE
      LREL(INDEXA) = NVID
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
         LENV(INDEXA. = 1
         LFOR(1,INDEXA) = MORF(1,NVID)
         LFOR(2,INDEXA) = MORF(2,NVID)
 3300   CONTINUE
 3400   CONTINUE
         NUSED = 0
C   CASE DATA LOOP
C      ALL SELECTED CASES USED IF NO RANDOM SELECT [INSAMP = 1]
         INSAMP = 1
         DO 4500 INDEXA = 1,NCASE
C   PICK NEXT CASE IN THIS GROUP.
         CALL USEMSK(INDEXA,$4450)
         IF(IRSX.NE.0) CALL SRAND(INSAMP)
         IF(INSAMP.LT.0) GO TO 4450
         NUSED = NUSED + 1
         ISSPNT = NUSED + IBSSPN
         ENCODE(8753,ISSP) ISSPNT
 8753   FORMAT(J4)
C   READ CASE DATA.
         READ(LIU) NC,(JCV(INDEX),INDEX = 1,NC),NH,(JMD(INDEX),
       1 INDEX = 1,NH),NT,(JTD(INDEX),INDEX = 1,NT)
         IDCASE = JMD(1)
         GO TO JUMPD,(3500,3870)
 3500   CONTINUE
C   MAKE "PRINT VARIABLE" ARRAY.
         DO 3850 INDEXB = 1,NSPEC
         DO 3550 INDEXC = 1,3
         LPRINT(INDEXC,INDEXB) = ' '
 3550   CONTINUE
C   BRANCH FOR BACKGROUND OR COMPUTED VARIABLE.
         IF(LREL(INDEXB).GT.0) GO TO 3650
         LOC = (-LREL(INDEXB)) - 1
         LEN = LENV(INDEXB)
         DO 3600 INDEXC = 1,LEN
         INDEXD = INDEXC + LOC
         IBUF(INDEXC) = JMD(INDEXD)
 3600   CONTINUE
         GO TO 3700
 3650   CONTINUE
         LOC = LREL(INDEXB)
         LEN = LENV(INDEXB)
         IBUF(1) = JCV(LOC)
 3700   CONTINUE
         IFOR(2) = LFOR(1,INDEXB)
         IFOR(3) = LFOR(2,INDEXB)
         ENCODE(IFOR,JBUF) (IBUF(INDEX),INDEX = 1,LEN)
         MOVE = MIN0(LEN,3)
         DO 3750 INDEXC = 1,MOVE
         LPRINT(INDEXC,INDEXB) = JBUF(INDEXC)
 3750   CONTINUE
```

LINE #   TEXT...

```
      IF(NDC(INDEXB).EQ.0) GO TO 3850
C  DECODE THIS VALUE.
      NDCS = NDC(INDEXB)
      DO 3800 INDEXC = 1,NDCS
      IF(LPRINT(1,INDEXB).NE.KARV(1,INDEXC,INDEXB)) GO TO 3800
      LPRINT(1,INDEXB) = KARV(2,INDEXC,INDEXB)
      LPRINT(2,INDEXB) = KARV(3,INDEXC,INDEXB)
      LPRINT(3,INDEXB) = KARV(4,INDEXC,INDEXB)
      GO TO 3850
3800  CONTINUE
3850  CONTINUE
3870  CONTINUE
C  WRITE PRINT AND TASK INFO. TO DIRECT ACCESS FILE.
      FIND(LDI'INDEXA)
      NREK = INDEXA
      LUNT = LDI
      WRITE(LDI'INDEXA,ERR=7300) ((LPRINT(INDEX,INDEX1),INDEX = 1,3),
     1 INDEX1 = 1,NSPEC),ISSP,IDCASE,NT,(JTD(INDEX),INDEX = 1,NT)
      GO TO JUMPE,(3900,4000,4100)
3900  CONTINUE
C  SEQUENTIAL OR RANDOM
      KASE(NUSED) = INDEXA
      KVIS(NUSED) = JHD(-IVRL)
      GO TO 4500
4000  CONTINUE
C  MINOR SORT FIELD ENTRY
      IF(MIRL.GT.0) GO TO 4050
      ISORT(2) = JHD(-MIRL)
      GO TO 4150
4050  CONTINUE
      ISORT(2) = JCV(MIRL)
      GO TO 4150
4100  CONTINUE
C  MAJOR SORT FIELD ENTRY
      ISORT(2) = ' '
4150  CONTINUE
      IF(MARL.GT.0) GO TO 4200
      ISORT(1) = JHD(-MARL)
      GO TO 4250
4200  CONTINUE
      ISORT(1) = JCV(MARL)
4250  CONTINUE
      ISORT(3) = INDEXA
      ISORT(4) = JHD(-IVRL)
      LOAR(NUSED) = NUSED
C  WRITE SORT FILE TO DIRECT ACCESS FILE.
      FIND(LDO'NUSED)
      NREK = NUSED
      LUNT = LDO
      WRITE(LDO'NUSED,ERR=7300) ISORT
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      GO TO 4500
 4450 CONTINUE
C  SKIP CASE NOT IN THIS GROUP.
      READ(LIO) MASH
 4500 CONTINUE
      IF(IPASS.NE.2) GO TO 5000
C  ON FIRST PASS, FETCH TASK TITLES AND LOCATOR TABLE.
      IPASS = 3
      READ(LIO) NULL,KHART
C  VERIFY TAPE POSITION.
      IF(KHART.NE.'END OF') GO TO 4900
      REWIND LD3
 4600 CONTINUE
      READ(LIO)  NWRD,(KTIT(INDEX),INDEX = 1,NWRD)
      WRITE(LD3) NWRD,(KTIT(INDEX),INDEX = 1,NWRD)
      IF(KTIT(1).NE.'END OF') GO TO 4600
      BACKSPACE LIO
      BACKSPACE LIO
      BACKSPACE LIO
      READ(LIO) NWOR,(LOCTSK(INDEX),INDEX = 1,NWOR)
      REWIND LD3
      IREC = 0
      LUNT = LD2
      DO 4800 INDEXA = 1,NTASK
      JREC = FLD( 0,18,LOCTSK(INDEXA))
      IF(IREC.EQ.JREC) GO TO 4700
      READ(LD3) NWRD,(KTIT(INDEX),INDEX = 1,NWRD)
      IREC = JREC
 4700 CONTINUE
      INCW = FLD(18,18,LOCTSK(INDEXA))
      INLW = FLD( 0,18,KTIT(INCW))
      LENT = INLW - INCW
      INCW = INCW + 1
      LOCTSK(INDEXA) = LENT / 11
      FIND(LD2'INDEXA)
      NREK = INDEXA
      WRITE(LD2'INDEXA,ERR=7300) (KTIT(INDEX),INDEX = INCW,INLW)
 4800 CONTINUE
      GO TO 5000
 4900 CONTINUE
      WRITE(6,8240) NCASE
 8240 FORMAT('1   "END OF DATA" NOT FOUND AFTER ',I6,' CASES.'/4X,
     1 'JOBIND ABORTS.')
      CALL ERTRAN(2)
 5000 CONTINUE
      REWIND LIO
      REWIND LD3
      GO TO JUMPF,(6000,5050,5100)
 5050 CONTINUE
C  RANDOM REPORT ORDER
```

LINE #   TEXT...

```
      CALL RANSEQ(NUSED,KASE)
      GO TO 6000
 5100 CONTINUE
C  ASCENDING SORT REPORT ORDER
      LUNT = LDO
      GO TO JUMPFA,(5150,5350)
 5150 CONTINUE
C  PERFORM MINOR SORT.
      DO 5300 INDEXA = 2,NUSED
      I = INDEXA - 1
      J = INDEXA
      J1 = LOAR(J)
      NREK = J1
      READ(LDO'J1,ERR=7400) JA,JB
 5200 CONTINUE
      I1 = LOAR(I)
      NREK = I1
      READ(LDO'I1,ERR=7400) IA,IB
      IF(FLD( 0,18,JB).GT.FLD( 0,18,IB)) GO TO 5300
      IF(FLD( 0,18,JB).LT.FLD( 0,18,IB)) GO TO 5250
      IF(FLD(18,18,JB).GT.FLD(18,18,IB)) GO TO 5300
      IF(FLD(18,18,JB).LT.FLD(18,18,IB)) GO TO 5250
      GO TO 5300
 5250 CONTINUE
      LINGER = LOAR(J)
      LOAR(J) = LOAR(I)
      LOAR(I) = LINGER
      J = I
      I = I - 1
      IF(I.GT.0) GO TO 5200
 5300 CONTINUE
 5350 CONTINUE
C  PERFORM MAJOR SORT.
      DO 5500 INDEXA = 2,NUSED
      I = INDEXA - 1
      J = INDEXA
      J1 = LOAR(J)
      NREK = J1
      READ(LDO'J1,ERR=7400) JA
 5400 CONTINUE
      I1 = LOAR(I)
      NREK = I1
      READ(LDO'I1,ERR=7400) IA
      IF(FLD( 0,18,JA).GT.FLD( 0,18,IA)) GO TO 5500
      IF(FLD( 0,18,JA).LT.FLD( 0,18,IA)) GO TO 5450
      IF(FLD(18,18,JA).GT.FLD(18,18,IA)) GO TO 5500
      IF(FLD(18,18,JA).LT.FLD(18,18,IA)) GO TO 5450
      GO TO 5500
 5450 CONTINUE
      LINGER = LOAR(J)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      LOAR(J) = LOAR(I)
      LOAR(I) = LINGER
      J = I
      I = I - 1
      IF(I.GT.0) GO TO 5400
 5500 CONTINUE
 6000 CONTINUE
C
C   GENERATE REPORT(S) FOR EACH CASE USED.
C
      DO 6600 INDEXA = 1,NUSED
C   GET NEXT CASE NUUMBER.
      GO TO JUMPF1,(6020,6040)
 6020 CONTINUE
      ICASE = KASE(INDEXA)
      GO TO 6100
 6040 CONTINUE
      NREK = LOAR(INDEXA)
      LUNT = LDO
      READ(LDO'NREK,ERR=7400) ISORT
      ICASE = ISORT(3)
 6100 CONTINUE
C   GET NEXT CASE.
      NREK = ICASE
      LUNT = LDI
      READ(LDI'ICASE,ERR=7400) ((LPRINT(INDEX,INDEX1),INDEX = 1,3),
     1 INDEX1 = 1,NSPEC),ISSP,IDCASE,NT,(JTD(INDEX),INDEX = 1,NT)
C   SORT TASKS.
      CALL H2SORT(NT,JTD)
C   WRITE REPORT TO PRINT SCRATCH FILE.
      NPAGE = 1
      IJOB(INDEXA) = IDCASE
      ISP(INDEXA) = ISSP
      ISSPN = ISSP
      IF(IX.NE.' ') ISSPN = IDCASE
      WRITE(LD3,8700) ITL,IDH,NPAGE
 8700 FORMAT(1H1,7A6,35X,A6,' PAGE ',I3/)
      GO TO JUMPG,(6110,6120)
 6110 CONTINUE
      WRITE(LD3,8720)((NIGH(INDEX,INDEX1),INDEX = 1,14),INDEX1 = 1,NHCS)
 8720 FORMAT(1X,14A6)
 6120 CONTINUE
      WRITE(LD3,8740)
 8740 FORMAT(1H )
      IF(NLSSP.NE.0) WRITE(LD3,8750) ISSPN
 8750 FORMAT(20X,'*********',/,' JOB NUMBER    =     * ',A4,' *',/,
     1,20X,'*********',/)
      GO TO JUMPH,(6130,6150)
 6130 CONTINUE
      DO 6140 INDEXC = 1,NSPEC
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      WRITE(LD3,8760) (ITLV(INDEX,INDEXC),INDEX = 1,3),
     1 (LPRINT(INDEX,INDEXC),INDEX = 1,3)
8760  FORMAT(1X,3A6,1X,3A6)
6140  CONTINUE
6150  CONTINUE
      WRITE(LD3,8780) LHDT
8780  FORMAT(1H0,72X,'PER CENT   CUMULATIVE'/1X,A5,8X,
     1 'TASK TITLE',48X,'TIME SPENT   PER CENT'/)
      LINES = 2 + NHCS + 1 + NSPEC + 4 + NLSSP
      GO TO JUMPI,(6160,6170)
6160  CONTINUE
C  TASK CUTOFF
      NT = MIN0(NT,KUTOFF)
6170  CONTINUE
      KPCT = 0
      LUNT = LD2
      DO 6250 INDEXC = 1,NT
      DO 6180 INDEXD = 1,99
      LTIA(INDEXD) = ' '
6180  CONTINUE
      KSAT = FLD( 0,18,JTD(INDEXC))
      LAVT = FLD(18,18,JTD(INDEXC))
      LINT = LOCTSK(KSAT)
      NREK = KSAT
      READ(LD2'KSAT,ERR=7400) ((LTIT(INDEX,INDEX1),INDEX = 1,11),
     1 INDEX1 = 1,LINT)
      LINES = LINES + LINT
      IF(LINES.LT.55) GO TO 6186
      GO TO JUMPL,(6182,6184)
6182  CONTINUE
C  PAGE CUTOFF
      IF(NPAGE.GE.KUTOFF) GO TO 6260
6184  CONTINUE
      NPAGE = NPAGE + 1
      WRITE(LD3,8700) ITL,IDR,NPAGE
      LINES = 2
6186  CONTINUE
      GO TO JUMPJ,(6190,6200)
6190  CONTINUE
      ENCODE(8800,LTIT(1,1)) INDEXC
8800  FORMAT(I5,1X)
6200  CONTINUE
      TVAL = LAVT
      KPCT = KPCT + LAVT
      CPCT = KPCT
      TVAL = TVAL/1000.
      CPCT = CPCT/1000.
      WRITE(LD3,8820) (LTIT(INDEX,1),INDEX = 1,11),TVAL,CPCT
8820  FORMAT(1X,11A6,5X,F7.2,5X,F7.2)
      IF(LINT.EQ.1) GO TO 6205
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      WRITE(LD3,8840)((LTIT(INDEX,INDEX1),INDEX = 1,11),INDEX1 = 2,LINT)
 8840 FORMAT(1X,11A6)
 6205 CONTINUE
      GO TO JUMPK,(6210,6250)
 6210 CONTINUE
C  CUMULATIVE PER CENT CUTOFF
      IF(KPCT.GT.KUTOFF) GO TO 6260
 6250 CONTINUE
 6260 CONTINUE
      GO TO JUMPM,(6270,6600)
 6270 CONTINUE
C  WRITE REPORT TO REPORT FILE.
      NPAGE = 1
      WRITE(LOU,8700) ITL,IDR,NAME
      GO TO JUMPN,(6280,6290)
 6280 CONTINUE
      WRITE(LOU,8720) ((NIGH(INDEX,INDEX1),INDEX = 1,14),
     1 INDEX1 = 1,NHCS)
 6290 CONTINUE
      WRITE(LOO,8740)
      IF(NLSSP.NE.0) WRITE(LD3,8750) ISSPN
      GO TO JUMPO,(6300,6320)
 6300 CONTINUE
      DO 6310 INDEXB = 1,NSPEC
      WRITE(LOU,8760) (ITLV(INDEX,INDEXB),INDEX = 1,3),
     1 (LPRINT(INDEX,INDEXB),INDEX = 1,3)
 6310 CONTINUE
 6320 CONTINUE
      WRITE(LOO,8780) LHDT
      LINES = 2 + NHCS + 1 + NSPEC + 4 + NLSSP
      GO TO JUMPP,(6330,6340)
 6330 CONTINUE
      NT = KUTOFF
 6340 CONTINUE
      KPCT = 0
      LUNT = LD2
      DO 6420 INDEXB = 1,NT
      DO 6350 INDEXC = 1,99
      LTIA(INDEXC) = ' '
 6350 CONTINUE
      KSAT = FLD( 0,18,JTD(INDEXB))
      LAVT = FLD(18,18,JTD(INDEXB))
      LINT = LOCTSK(KSAT)
      NREK = KSAT
      READ(LD2'KSAT,ERR=7400) ((LTIT(INDEX,INDEX1),INDEX = 1,11),
     1 INDEX1 = 1,LINT)
      LINES = LINES + LINT
      IF(LINES.LT.55) GO TO 6356
      GO TO JUMPS,(6352,6354)
 6352 CONTINUE
```

LINE #   TEXT...

```
      IF(NPAGE.GE.KUTOFF) GO TO 6430
 6354 CONTINUE
      NPAGE = NPAGE + 1
      WRITE(LOU,8700) ITL,IDR,NPAGE
      LINES = 2
 6356 CONTINUE
      GO TO JUMPQ,(6360,6370)
 6360 CONTINUE
      ENCODE(8800,LTIT(1,1)) INDEXB
 6370 CONTINUE
      TVAL = LAVT
      KPCT = KPCT + LAVT
      CPCT = KPCT
      TVAL = TVAL/1000.
      CPCT = CPCT/1000.
      WRITE(LOO,8820) (LTIT(INDEX,1),INDEX = 1,11),TVAL,CPCT
      IF(LINT.EQ.1) GO TO 6375
      WRITE(LOO,8840)((LTIT(INDEX,INDEX1),INDEX = 1,11),INDEX1 = 2,LINT)
 6375 CONTINUE
      GO TO JUMPR,(6380,6420)
 6380 CONTINUE
      IF(KPCT.GT.KUTOFF) GO TO 6430
 6420 CONTINUE
 6430 CONTINUE
      GO TO JUMPT,(6500,6530)
 6500 CONTINUE
      KFLD = KVIS(INDEXA)
      GO TO 6560
 6530 CONTINUE
      KFLD = ISORT(4)
 6560 CONTINUE
C  CLOSE THIS REPORT.
      CALL REPENI(IS4,IDR,ITL,KFLD,JSEQR,ISSPN)
 6600 CONTINUE
      IF(ITRACE.NE.1) GO TO 6432
      IF(NUSED.GE.201) WRITE(6,8755) IDSORT,NUSED
 8755 FORMAT(' TRACE IGNORED ON SORT #',I3,' BECAUSE NUSED =',I5)
      IF(NUSED.GE.201) GO TO 6432
      N = NUSED
      WRITE(ISU) (IALLV(I),I=1,13),N,(IJOB(I),I=1,N),N,(ISP(I),I=1,N)
 6432 CONTINUE
C  PRINT REPORT(S).
      ENDFILE LD3
      CALL ERTRAN(6,IUSE24)
      DO 6800 INDEXA = 1,NCOP
      REWIND LD3
      CALL REWINM
 6700 CONTINUE
      CALL DREAD(NWDS,NIRP,ORANGE,$6800,$6800)
      WRITE(6,8999) (NIRP(INDEX),INDEX = 1,NWDS)
```

LINE #   TEXT...

```
8999  FORMAT(22A6)
      GO TO 6700
6800  CONTINUE
      GO TO 100
C  PROGRAM CLOSEOUT
7000  CONTINUE
      CALL MARGIN
      WRITE(6,8900) IDS1,IDS2,NAME
8900  FORMAT('    JOBIND COMPLETE FOR STUDY ',A6,A2,' . '/4X,A6,
     1 ' ENDS.')
      IF(ITRACE.EQ.1) ENDFILE ISU
      STOP
C
C  ERROR ROUTINES
C
7100  CONTINUE
      READ(0,8180) LWD
      WRITE(6,8920) LWD
8920  FORMAT('1   CONTROL CARD ERROR ON "',A3,'" CARD.  SEARCHING FOR "E
     1ND" CARD.')
7150  CONTINUE
      READ(5,8180,END=7000) LWD
      IF(LWD.NE.'END') GO TO 7150
      GO TO 100
7200  CONTINUE
      WRITE(6,8940) NVID
8940  FORMAT('1    ',A3,' IS INVALID VARIABLE SPECIFICATION.  SKIPPING TO
     1 NEXT REPORT.')
      GO TO 100
7300  CONTINUE
      WRITE(6,8960) NREK,LUNT
8960  FORMAT('1   DIRECT ACCESS WRITE ERROR ON RECORD',I6,' ,UNIT',
     1 I3,' .  JOBIND ABORTS.')
      CALL ERTRAN(2)
7400  CONTINUE
      WRITE(6,8980) NREK,LUNT
8980  FORMAT('1   DIRECT ACCESS READ ERROR ON RECORD',I6,' ,UNIT',
     1 I3,' .  JOBIND ABORTS.')
      CALL ERTRAN(2)
7600  CONTINUE
      WRITE(6,8990) (LDICT(INDEX,INDEXB),INDEX = 1,15)
8990  FORMAT('1   FORMAT ERROR IN DICTIONARY ENTRY :'/4X,15A6/4X,'JOBIND
     1 ABORTS.')
      CALL ERTRAN(2)
C
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

REPEN1 CLOSE REPORTS & PUNCH SPECIAL JOBIND REQUEST CARD

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE REPEN1(ISTUDY,IDREPT,LABEL,IVX,JOS,ISSPN)
      DIMENSION LABEL(7)
      REWIND 2
      READ(2) JSEQ
      WRITE(1,100) ISTUDY, IDREPT, LABEL, IVX, JOS, ISSPN, JSEQ
100   FORMAT(A4,A6,7A6,1X,A6,7X,I4,1X,A4,1X,I4)
      JSEQ = JSEQ + 1
      WRITE(26,101) JSEQ
101   FORMAT('BEGIN REPORT',A6)
      REWIND 2
      WRITE(2) JSEQ
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      INTEGER BLANK,TIDUTY,TITLES
      DIMENSION AA(2)
      EQUIVALENCE (AA(1),JA(1))
       DIMENSION JA(50),STDY(2),ITEMP(2),ITEST(2),LTAB(2027),KTSK(27),
     1 TIDUTY(26,22),TITLES(2,605),KTIT(465),HEADNG(14)
      DIMENSION LABEL(8)
      DATA ITEST,MBIT/'END OF DATA',262144/
      DATA BLANK/6H      /
       DATA SPID/6HJOBINV/
       DATA LABEL/48HJOB INVENTORY(DUTY/TASK TITLES)
       CALL SETUP(10,26,0,0,0,0,20,0,0,0,0,JSEQ)
       IST=20
       IMT=10
       LT=26
      LR=5
      LP=6
      LPAGE=1
      NLINE=0
      LINMAX=54
      ILM=0
      ILM1=0
      REWIND IMT
C
C     READ COMMUNICATION REGION
C
      READ(IMT) K,(JA(L),L=1,K)
      NDUTY=JA(9)
      NTASK=JA(12)
      READ(LR,100) PROG,STDY(1),STDY(2),NTCARD,IBCD
      IBCD = 1 -IBCD
      IF(IBCD.LT.0) IBCD = 1
  100 FORMAT(A6,1X,A6,A2,2I2)
    2 IF(STDY(1).EQ.AA(1).AND.STDY(2).EQ.AA(2)) GO TO 6
      WRITE(LP,102) STDY(1),STDY(2),JA(1),JA(2)
  102 FORMAT(43H0CONTROL CARD ERROR. STUDY IDENTIFICATION 'A6,A2,
     1       13H' SHOULD BE ',A6,A2,26H' AS ON HISTORY DATA TAPE.)
      CALL ERTRAN(2)
    6 REWIND IST
      WRITE(LP,106) LABEL,SPID,LPAGE
      IF(IBCD.NE.0) WRITE(LT,106) LABEL,SPID,LPAGE
      WRITE(LP,107)
      IF(IBCD.NE.0) WRITE(LT,107)
  107 FORMAT(  54X,23HUNITED STATES AIR FORCE,/59X,13HJOB INVENTORY,//)
      IF(NTCARD.EQ.0) GO TO 8
      DO 4 I=1,NTCARD
      READ(LR,104) HEADNG
  104 FORMAT(13A6,A2)
      IF(IBCD.NE.0) WRITE(LT,105) HEADNG
    4 WRITE(LP,105) HEADNG
  105 FORMAT(1H ,13A6,A2)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      WRITE(LP,108)
      IF(IBCD.NE.0) WRITE(LT,108)
  108 FORMAT(1X,/)
    8 READ(IMT) K,ITEMP(1),ITEMP(2)
      IF(ITEMP(1).NE.ITEST(1).AND.ITEMP(2).NE.ITEST(2)) GO TO 8
   10 READ(IMT) K,(LTAB(L),L=1,K)
      WRITE(IST) K,(LTAB(L),L=1,K)
      IF(LTAB(1).NE.ITEST(1).AND.LTAB(2).NE.ITEST(2)) GO TO 10
      BACKSPACE IST
      BACKSPACE IST
      BACKSPACE IST
C
C     READ TITLE-LOCATOR TABLE, LOCATE AND READ DUTY TITLES
C
      DO 15 I=1,26
      DO 15 J=1,22
   15 TIDUTY(I,J)=BLANK
      READ(IST) K,(LTAB(L),L=1,K)
      READ(IST) K,(KTSK(L),L=1,K)
      REWIND IST
      IB=NTASK+1
      IE=NTASK+NDUTY
      LM=LTAB(IB)/MBIT
      DO 29 I=1,LM
   29 READ(IST) K,(KTIT(L),L=1,K)
      ILM=LM
      KNT=0
      DO 30 I=IB,IE
      LM=LTAB(I)/MBIT
      LN=MOD(LTAB(I),MBIT)
      IF(ILM.NE.LM) READ(IST) K,(KTIT(L),L=1,K)
      ILM=LM
      LM=MOD(KTIT(LN),MBIT)
      LK=KTIT(LN)/MBIT
      NW=LK-LM
      IF(NW.GT.22) NW=22
      KNT=KNT+1
      DO 20 J=1,NW
      LMJ=LM+J
   20 TIDUTY(KNT,J)=KTIT(LMJ)
       DECODE(200,TIDUTY(KNT,1)) X, Y, Z
  200  FORMAT(A1,A1,A4)
       ENCODE(200,TIDUTY(KNT,1)) BLANK,X,Z
   30 CONTINUE
C
C     LOCATE AND READ TASKS FOR EACH DUTY
C
      REWIND IST
      II=1
      LINE=-1
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      LINEMX=LINMAX-8
      JJ=1
      ILM=0
      DO 38 I=1,2
      DO 38 L=1,605
   38 TITLES(I,L)=BLANK
      DO 60 I=1,NDUTY
      JUMP=1
      JB=0
      LINE=LINE+1
   39 IB=MOD(LINE,LINEMX)*11+1
      IE=IB+10
      IF(LINE.GE.LINEMX)II=2
      LINE=LINE+1
      DO 40 J=IB,IE
      JB=JB+1
   40 TITLES(II,J)=TIDUTY(I,JB)
      IF(TIDUTY(I,JB+2).EQ.BLANK.OR.JUMP.EQ.2) GO TO 42
      JUMP=2
      GO TO 39
   42 CONTINUE
      LINE=LINE+1
      LM=KTSK(I)/MBIT
      LN=MOD(KTSK(I),MBIT)
      DO 55 LL=LM,LN
      IPRINT=1
      LMI=LTAB(LL)/MBIT
      LNI=MOD(LTAB(LL),MBIT)
      IF(ILM.NE.LMI) READ(IST) K,(KTIT(L),L=1,K)
      ILM=LMI
      LMI=MOD(KTIT(LNI),MBIT)
      LKI=KTIT(LNI)/MBIT
      NW=LKI-LMI
      NOL=NW/11
      IF(LINE.GE.LINEMX) II=2
      IF(LINE.GE.2*LINEMX) IPRINT=2
      GO TO (44,51),IPRINT
   44 CONTINUE
      IPRINT=1
      LMJ=LMI
      DO 55 III=1,NOL
      IF(LINE.GE.LINEMX)II=2
      IB=MOD(LINE,LINEMX)*11+1
      IE=IB+10
      LINE=LINE+1
      DO 52 JJ=IB,IE
      LMJ=LMJ+1
   52 TITLES(II,JJ)=KTIT(LMJ)
      DECODE(200,TITLES(II,IB)) X, Y, Z
      ENCODE(200,TITLES(II,IB)) BLANK, X, Z
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      IF(LL.NE.LN.OR.III.NE.NOL) GO TC 50
      LX = LINE + 3
      IF(LX.GT.2*LINEMX) GO TO 51
      IF(I.EQ.NDUTY) GO TO 51
 50   IF(LINE.LT.2*LINEMX) GO TO 55
   51 IB=1
      IE=11
      IF(LPAGE.EQ.1) GO TO 49
      WRITE(LP,106) LABEL, SPID, LPAGE
      IF(IBCD.NE.0) WRITE(LT,106) LABEL,SPID,LPAGE
 106  FORMAT( '1',8A6,29X,A6,' PAGE ',I3,//)
   49 CONTINUE
      DO 53  J=1,LINEMX
      WRITE(LP,103) (TITLES(1,JJ),JJ=IB,IE),(TITLES(2,JJ),JJ=IB,IE)
      IF(IBCD.NE.0)
     1WRITE(LT,103) (TITLES(1,JJ),JJ=IB,IE),(TITLES(2,JJ),JJ=IB,IE)
      IB=IB+11
   53 IE=IE+11
      II=1
      LINE=0
      LINEMX=LINMAX-2
      LPAGE=LPAGE+1
      DO 54 J=1,2
      DO 54 JJ=1,605
   54 TITLES( J,JJ)=BLANK
      GO TO (55,44),IPRINT
   55 CONTINUE
   60 CONTINUE
  103 FORMAT(22A6)
C
C     PUNCH REQUEST CARD IF REPORT IS ADDED TO BCD REPORT TAPE
C
       IF(IBCD.NE.0) CALL REPEND(STDY(1),SPID,LABEL,JSEQ)
      CALL MARGIN
      STOP
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      DIMENSION JA(50), KOS(1000), KDUTY(26), ARRAY(7900)
      DIMENSION KB(1726), KC(1000), KD(1000), LABEL(42)
      COMMON JA, KOS, KB, KDUTY, LABEL, KC, KD, IS, JSEQ
      DATA IDPRG/'JOBSPC'/
      EQUIVALENCE (JA(3),NCASES),(JA(12),NTASK), (JA(9),NDUTY)
      CALL SETUP9(10,25,26,0,0,0,20,21,22,0,0,JSEQ)
C  READ CONTROL CARD
      READ(5,800) IDP, ID1, ID2, NTIT
  800 FORMAT(A6,1X,A6,A2,1X,I1)
      IF(IDP.EQ.IDPRG) GO TO 1
      WRITE(6,801) IDP
  801 FORMAT('1INCORRECT PROGRAM ID *',A6,'* - RUN HALTS')
      CALL ERTMAN(2)
    1 IPT = 10
      REWIND IPT
C  READ COMMUNICATION REGION
      READ(IPT) K, (JA(I),I=1,K)
      IF(ID1.EQ.JA(1).AND.ID2.EQ.JA(2))GO TO 2
      WRITE(6,802) JA(1), JA(2), ID1, ID2
  802 FORMAT('1TAPE-ID MISMATCH, TAPE = ',A6,A2,', CARD = ',A6,A2,'.')
      STOP
    2 CONTINUE
      IS = 1
C
C  READ TITLE CARDS IF ANY
      NWORDS = 14 * NTIT
      IF(NTIT.EQ.0) GO TO 3
      READ(5,803) (LABEL(I),I=1,NWORDS)
  803 FORMAT(13A6,A2)
    3 CONTINUE
C
C  READ RELATIVE LOCATION TABLES
C
      READ(IPT) K, (KOS(L),L=1,K), M, (KDUTY(L),L=1,M)
C
C
      NTD = NTASK + NDUTY
      CALL DISK(NTD,99)
      NTD2 = NTD + NTD
      NC35 = (NCASES - 1) / 35 + 1
      NAVAIL = 7900
      MAXJD = (NAVAIL - NTD2) / (100 + NTD2 + NC35)
C
C  STORE INTO COMMUNICATIONS REGION
C
      JA(19) = NTIT
      JA(20) = NWORDS
      JA(45) = NTD
      JA(48) = MAXJD
      JA(49) = NC35
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
C  ESTABLISH STARTING LOCATIONS FOR JOBSPI ARRAYS
C
      NX1=NTD+1
      NX2=NX1+NTD
      NX3=NX2+NTD*MAXJD
      NX4=NX3+NTD*MAXJD
      NX5=NX4+NC35*MAXJD
      NX6=NX5+MAXJD
      NX7=NX6+MAXJD
      NX8=NX7+9*MAXJD
      NX9=NX8+14*MAXJD
      NX10=NX9+9*MAXJD
      NX11=NX10+36*MAXJD
      NX12=NX11+8*MAXJD
      NX13=NX12+9*MAXJD
      NX14=NX13+MAXJD
      NX15=NX1
      LNTHBF=NAVAIL-NTD
C
C***** SPACE PAST HISTORY VARIABLE DICTIONARY
C
      K = JA(4)
      DO 5 LMK =1,K
      READ (IPT) I
    5 CONTINUE
C
      CALL JOBSPI(MAXJD,NTD,NC35,ARRAY,ARRAY(NX1),ARRAY(NX2),ARRAY(NX3),
     1 ARRAY(NX4),ARRAY(NX5),ARRAY(NX6),ARRAY(NX7),ARRAY(NX8),ARRAY(NX9)
     2,ARRAY(NX10),ARRAY(NX11),ARRAY(NX12),ARRAY(NX13),ARRAY(NX14),
     3 ARRAY(NX15),LNTHBF)
      CALL MARGIN
      STOP
      SUBROUTINE JOBSPI(MAXJD,NTD,NC35,NDTLT,AKOS,AVE,ANUMP,NMASK,KSWT,
     1 MBRS,INPT,IHEAD,IVL,IVAR,IOPN,ISGT,NNN,JDOK,MBUF,LNTHBF)
      COMMON JA, KOS, KB, KDUTY, LABEL, KE, KD, IS, JSEQ
      DIMENSION JA(50), KOS(1000), KDUTY(26), KB(1726), LABEL(42),
     1KE(1000), KD(1000), IEND(2), MBUF(1), KTST(9), FJA(1),
     2LBUF(54), IBUF(2,2), FBUF(2,2), LIST(2,1000)
      DIMENSION NDTLT(NTD),AKOS(NTD),AVE(NTD,MAXJD),ANUMP(NTD,MAXJD),
     1NMASK(NC35,MAXJD),KSWT(MAXJD),MBRS(MAXJD),INPT(9,MAXJD),
     2IHEAD(14,MAXJD),IVL(9,MAXJD),IVAR(36,MAXJD),IOPN(8,MAXJD),
     3ISGT(9,MAXJD),NNN(MAXJD),JDOK(MAXJD)
      EQUIVALENCE (FJA,JA), (F21,I21), (IBUF,FBUF),
     1(JA(3),NCASES),(JA(9),NDUTY),(JA(12),NTASK)
      DATA IEND/'END OF DATA'/
      ASSIGN 701 TO IPASS
      DATA NPASS/0/
C  UNIT ASSIGNMENTS
      ITP = 10
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      IO = 25
      ITAP = 26
      MJD = 20
      INJ = 22
      NTI = 21
      J14 = 8
      MBIT = 2**18
      MAVAIL = LNTHBF - 7*NTD - 14 - NC35
      NX1 = NTD + 1
      NX2 = NTD + NX1
      NX3 = NTD + NX2
      NX4 = NX3 + 14
      NX5 = NX4 + NC35
      NX6 = NX5 + 4*NTD
      JDFLAG = 0
 8    NJD = 1
C
C
C***** NEW GROUP OF JOB DESCRIPTIONS
C***** JDFLAG=1, IF THIS IS LAST GROUP
C***** NPASS COUNTS GROUPS OF JOB DESCRIPTIONS
C***** NJD - NUMBER OF JOB DESCRIPTIONS IN THIS GROUP
C
      REWIND INJ
      NPASS=NPASS+1
      NDIM=MAXJD*NC35
      CALL ZEROI(NMASK,NDIM)
 20   READ(5,5,END=21) INPT(9,NJD), (IHEAD(L,NJD),L=1,8), JOBIND
 5    FORMAT(A6,1X,8A6,1X,A1)
C
C***** NEW JOB DESCRIPTION
C***** MBRS - NUMBER OF MEMBERS IN EACH JOB DESCRIPTION
C***** JDOK=1 IMPLIES ERROR IN A JOB DESCRIPTION
C
      MBRS(NJD)=0
      JDOK(NJD)=0
       IERR = 0
C
      IF(JOBIND.NE.'1') GO TO 745
      READ(5,740) ITYPE, ISUB, NWORDS
 740  FORMAT(3X,A1,A3,1X,A1)
      CALL BCDBIN(ISUB,ISUB,3,$742)
      IF(NWORDS.NE.'2') NWORDS = '1'
      CALL BCDBIN(NWORDS,NWORDS,1,$742)
      IF(ITYPE.EQ.'V') GO TO 741
      WRITE(6,777)
 742  WRITE(6,778)
 777  FORMAT(' ** EXACT MATCHES FOR V VARIABLES ONLY')
 778  FORMAT(' ** INVALID ENTRY ABOVE. REPORT SKIPPED')
      JDOK(NJD) = 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      WRITE(6,511) INPT(9,NJD), (IHEAD(L,NJD),L=1,8)
757   CONTINUE
      READ(5,744,END=17) IFEND
      IF(IFEND.NE.'END') GO TO 757
      GO TO 17
741   ISGT(1,NJD) = MOD(KOS(ISUB),MBIT)
      NLIST = 0
743   NLIST = NLIST + 1
      READ(5,744) (LIST(I,NLIST),I=1,NWORDS)
744   FORMAT(2A6)
      IF(LIST(1,NLIST).NE.'END') GO TO 743
      NLIST = NLIST - 1
      NNN(NJD) = -NLIST
      WRITE(INJ) NJD,NWORDS,NLIST,((LIST(I,J),I=1,NWORDS),J=1,NLIST)
      GO TO 746
745   CONTINUE
C
C***** READ VARIABLE NAMES AND OPERATORS
C
      IPRNT = 0
      CALL SAMPLE(NJD,ISGT(1,NJD),NN,IERR,IPRNT)
      NNN(NJD)=NN
       N = 1
       DO 16 K =1,NN
      IF(ISGT(K,NJD).GE.4)GO TO 41
      IF(ISGT(K,NJD).GE.0)GO TO 40
C
C***** READ LIMITS - ALPHANUMERIC
C
      READ 15,L,KTST(K),((IBUF(J,M),M=1,L),J=1,2)
   15 FORMAT(I2,2X,I3,1X,4A6)
      KTST(K)=-KTST(K)
      GO TO 42
C
C***** READ LIMITS - INTEGER
C
40    READ(5,18) KTST(K), IBUF(1,1), IBUF(2,1)
18    FORMAT(4X,I3,1X,I6,6X,I6)
      L = 1
      GO TO 42
C
C***** READ LIMITS - FLOATING POINT
C
41    READ(5,19) KTST(K), FBUF(1,1), FBUF(2,1)
19    FORMAT(4X,I3,1X,2F12.0)
      L = 1
   42 CONTINUE
        KL = L*2
      IF(ISGT(K,NJD).NE.KTST(K))IERR=IERR+1
       DO 30 M=1,L
```

LINE #   TEXT...

```
          J=N+M-1
          IVAR(J,NJD)=IBUF(1,M)
          J=J+L
       30 IVAR(J,NJD)=IBUF(2,M)
           N = N+KL
          IVL(K,NJD)=L
   16      CONTINUE
C
C***** READ DATA CARD FOR THIS JOB DESCRIPTION
C
746       CONTINUE
          READ(5,10) KSW, LIM1, LIM2, LIM3, LIM4, KSRT, IBCD
          IBIN = IBCD
          ISPEC = 0
10        FORMAT(I1,1X,I5,1X,I5,1X,I4,1X,I3,1X,I1,1X,I1,1X,I1,1X,I1)
          IF(KSRT.LE.1) GO TO 9010
9000      FORMAT(' INVALID SORT OPTION IN CC24. SET = 0 FOR ',A6)
          IDSP = INPT(9,NJD)
          WRITE(6,9000) IDSP
          KSRT = 0
9010      KSWT(NJD) = KSRT + 1
          IF(LIM1.EQ.0.AND.JOBIND.EQ.'1') LIM1 = 1
          IF(LIM1.EQ.0) LIM1 = 5
          IF(LIM2.EQ.0) LIM2 = NCASES
          IF(LIM3.EQ.0) LIM3 = NTASK
          IF(LIM4.EQ.0) LIM4 = 100
          IF(IBCD.EQ.0) IBCD = 2
          IF(IBIN.EQ.0) IBIN = 2
          IF(IBCD.LE.2.AND.IBIN.LE.2) GO TO 4
          IBCD = 1
          IBIN = 1
          WRITE(6,3)
3         FORMAT(' INVALID BINARY/BCD TAPE OPTIONS - DESCRIPTION SKIPPED.')
          IERR = 1
4         IF(ISPEC.GT.1) IERR = 1
          IF(KSW.EQ.0) KSW = 3
          INPT(1,NJD) = KSW
          INPT(2,NJD) = LIM1
          INPT(3,NJD) = LIM2
          INPT(4,NJD) = LIM3
          INPT(5,NJD) = LIM4
          INPT(6,NJD) = IBCD
          INPT(7,NJD) = IBIN
          INPT(8,NJD) = ISPEC
C
          IF( KSW.GT.0.AND.KSW.LE.3) GO TO 12
          WRITE(6,11)
11        FORMAT(' INVALID DESCRIPTION OPTION IN CC1. SET = 3')
          KSW = 3
12        IF(IERR.EQ.0) GO TO 17
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      WRITE(6,511) INPT(9,NJD),(IHEAD(L,NJD),L=1,8)
511   FORMAT(' SKIPPING REPORT DUE TO ERRORS. ID = ',A6,2X,8A6)
      JDOK(NJD)=1
   17 CONTINUE
C
C***** TEST IF MAX. NO. OF J.D. HAS BEEN REACHED FOR THIS GROUP
C
      IF(NJD.GE.MAXJD)GO TO 22
      NJD=NJD+1
      GO TO 20
C
C***** END OF DATA, TEST IF ANY J.D. IN THIS GROUP
C
   21 IF(NJD.EQ.1)GO TO 60
      NJD=NJD-1
C
C***** INDICATE THAT THIS IS THE LAST GROUP
C
      JDFLAG=1
C
C***** PROCESS JOB DESCRIPTIONS
C
   22 REWIND NT1
      NDIM=NTO*MAXJD
      CALL ZERO(AVE,NDIM)
      CALL ZERO(ANUMP,NDIM)
      DO 490 KK=1,NCASES
C
C***** LOOP FOR EACH CASE
C
      GO TO IPASS, (701,702)
701   IF(NPASS.GT.1) GO TO 705
      READ(ITP) NC,(KE(I),I=1,NC),NH,(KD(I),I=1,NH),NT,(KB(I),I=1,NT)
      WRITE(NT1)NC,(KE(I),I=1,NC),NH,(KD(I),I=1,NH),NT,(KB(I),I=1,NT)
      GO TO 703
705   ASSIGN 702 TO IPASS
702   READ(NT1) NC,(KE(I),I=1,NC),NH,(KD(I),I=1,NH),NT,(KB(I),I=1,NT)
C
C***** NFULL=0 INDICATES THIS DATA HAS NOT YET BEEN EXPANDED
C
703   NFULL = 0
      DO 23 IJD=1,NJD
C
C***** LOOP FOR JOB DESCRIPTIONS IN THIS GROUP
C***** NN - NUMBER OF VARIABLES IN THIS J.D.
C
      NN=NNN(IJD)
      IF(NN.LT.0) GO TO 747
      IF(JDOK(IJD).NE.0)GO TO 23
      DO 24 I=1,36
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
 24 LBUF(I)=IVAR(I,IJD)
     N = 0
     DO 105 L = 1,NN
C
C***** LOOP FOR VARIABLES
C
      KL=ISGT(L,IJD)
C
C***** NEG. KL, INPUT VARIABLE            POSITIVE KL, COMPUTED VARIABLE
C
      IF(KL.GT.0)GO TO 33
      KL=-KL
C
C***** TEST VARIABLE LENGTH
C
                KM =     KOS(KL)/MBIT
      IF(KM.NE.IVL(L,IJD)) GO TO 550
C
  500 CONTINUE
C
C***** TEST IF INPUT VARIABLE IS IN RANGE
C
      KN = MOD(KOS(KL),MBIT)
      KLO = N + 1
      KHI = KLO + KM
      KTST(L) = LASCMP(KD(KN),LBUF(KLO),LBUF(KHI),KM)
      N = N + 2*KM
      GO TO 105
C
C  TEST COMPUTED VARIABLE RANGE
C
33    KTST(L) = 0
      IF(LBUF(N+1).LE.KE(KL).AND.LBUF(N+2).GE.KE(KL)) KTST(L) = 1
      N = N + 2
  105  CONTINUE
C
C***** PERFORM LOGICAL COMBINATION OF VARIABLES
C
      IERR = 0
      ITRA = 1
      CALL SELECT(IJD,KTST,$300,$550)
C***** ITRA=2 IMPLIES THIS CASE IS A MEMBER OF THIS J.D.
      GO TO 200
C
550   JDOK(IJD) = 1
      GO TO 23
C  OLD JOBIND TYPE SELECTION SECTION
747   REWIND INJ
748   READ(INJ) JJD, NWRD, NLIST, ((LIST(I,J),I=1,NWRD),J=1,NLIST)
      IF(JJD.NE.IJD) GO TO 748
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      ITRA = 1
      LOC = ISGT(1,IJD)
      LOC1 = LOC + 1
      DO 749 INDEX = 1, NLIST
      IF(KD(LOC).NE.LIST(1,INDEX)) GO TO 749
      ITRA = 2
749   CONTINUE
      GO TO (300,200), ITRA
750   CONTINUE
C
C***** SET APPROPRIATE NMASK BIT EQUAL TO ONE
C
 200    JPOINT = MOD(KK,35)
        JWORD  = KK/35 +1
       IF(JPOINT.NE.0)  GO TO  2015
        JPOINT = 35
        JWORD = JWORD - 1
 2015 NMASK(JWORD,IJD)=NMASK(JWORD,IJD)+2**(JPOINT-1)
C
C***** INCREMENT MEMBERS COUNTER
C
       MBRS(IJD)=MBRS(IJD)+1
       ITRA = 2
  300  CONTINUE
C
C***** TEST IF JOB DESCRIPTION IS OKAY
C
       IF(JDOK(IJD).NE.0)GO TO 27
C
C***** TEST IF THIS CASE IS A MEMBER OF THIS J.D.
C
       IF(ITRA.EQ.1)GO TO 27
C
C***** TEST IF DATA HAS BEEN EXPANDED
C
       IF(NFULL.NE.0)GO TO 31
C
C***** EXPAND CASE DATA - PREVIOUSLY SUBROUTINE FULL
C
       CALL ZERO(AKOS,NDUTY+NTASK)
       DO 201 L = 1, NT
       ITASK = KB(L) / MBIT
       AKOS(ITASK) = (FLOAT(MOD(KB(L),MBIT))) / 1000.0
   201 CONTINUE
       DO 301 L=1,NDUTY
       KQ=KDUTY(L)/MBIT
       KP=MOD(KDUTY(L),MBIT)
       KR=L+NTASK
       DO 401 KL=KQ,KP
       AKOS(KR)=AKOS(KR)+AKOS(KL)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
  401 CONTINUE
  301 CONTINUE
C
C***** SET INDICATOR SO DATA WILL NOT BE EXPANDED AGAIN
C
   31 NFULL=1
C
      DO 450 LL = 1, NTD
C
C***** LOOP FOR TASKS AND DUTIES
C***** TEST IF THIS MEMBER PARTICIPATED IN THIS TASK OR DUTY
C
            IF (AKOS(LL).EQ.0.0) GO TO 450
C
C***** INCREMENT NUMBER PARTICIPATING COUNTER
C
      ANUMP(LL,IJD)=ANUMP(LL,IJD)+1.0
C
C***** ADD CASE DATA TO JOB DESCRIPTION SUMMATION
C
      AVE(LL,IJD)=AVE(LL,IJD)+AKOS(LL)
450         CONTINUE
   27 CONTINUE
   23 CONTINUE
490         CONTINUE
      REWIND NT1
      IF(NPASS.GT.1) GO TO 700
C
C***** IF FIRST PASS, WRITE DUTY-TASK TITLES ON NT2 FOR JOBPR1
C
      READ(ITP) K, JEND,KEND
      IF(JEND.EQ.IEND(1).AND.KEND.EQ.IEND(2)) GO TO 720
      WRITE(6,718)
718   FORMAT('1FIRST END OF DATA DOES NOT FOLLOW CASE DATA - RUN HALTS')
      STOP
C
720   CONTINUE
      READ(ITP)  K,(KE(I),I=1,K)
      WRITE(MJD) K,(KE(I),I=1,K)
      IF(KE(1).NE.IEND(1).OR.KE(2).NE.IEND(2)) GO TO 720
      BACKSPACE ITP
      BACKSPACE ITP
      BACKSPACE ITP
      READ(ITP) K, (NDTLT(I),I=1,K)
      REWIND ITP
      REWIND MJD
      IREC = 0
      DO 722 I = 1, NTD
      JREC = NDTLT(I) / MBIT
      IF(IREC.EQ.JREC) GO TO 721
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      READ(MJD) K, (KE(J),J=1,K)
      IREC = JREC
  721 K = MOD(NDTLT(I),MBIT)
      LAST = KE(K) / MBIT
      IFST = MOD(KE(K),MBIT)
      I21 = LAST - IFST
      AKOS(I) = F21
      IFST = IFST + 1
      ITO = 1
      CALL DSKSTO(I21,KE(IFST),ITO,1)
  722 CONTINUE
      REWIND MJD
      WRITE(MJD) NTD, (AKOS(I),I=1,NTD)
      GO TO 712
  700 CONTINUE
      READ(MJD) ISKIP
  712 CONTINUE
      NOK=0
      DO 25 IJD=1,NJD
C
C***** LOOP FOR J.D. IN THE GROUP
C***** TEST IF J.D. IS OKAY
C
      IF(JDOK(IJD).NE.0)GO TO 25
C
C***** TEST IF NO. MEMBERS IS IN PRESCRIBED RANGE
C
      MIN=INPT(2,IJD)
      MAX=INPT(3,IJD)
      IF(MBRS(IJD).LT.MIN.OR.MBRS(IJD).GT.MAX)GO TO 501
C
C***** LOAD THE COMMUNICATION REGION FOR THIS J.D.
C
      JA(17)=MBRS(IJD)
      JA(24)=INPT(9,IJD)
      JA(25)=INPT(8,IJD)
      JA(26)=INPT(7,IJD)
      JA(27)=INPT(6,IJD)
     FJA(28)=INPT(5,IJD)
      JA(29)=INPT(4,IJD)
      JA(30)=MAX
      JA(31)=MIN
      JA(34)=INPT(1,IJD)
      KSRT=KSWT(IJD)
      JA(35) = 'JOBSPC'
      JA(36) = 1
      IF(NNN(IJD).GE.0) GO TO 751
      JA(35) = 'SAMSEL'
      JA(36) = 0
  751 CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C***** PRINT JOB DESCRIPTION
C
      J5=50
      N = NC35
      WRITE(MJD) (AVE(I,IJD),I=1,NTD),(ANUMP(I,IJD),I=1,NTD),KSRT,
     1(IHEAD(I,IJD),I=1,8),N,(NMASK(I,IJD),I=1,N),J5,(JA(I),I=1,J5)
      NOK=NOK+1
      GO TO 25
  501 WRITE(6   ,502)MBRS(IJD),MIN,MAX
  502 FORMAT(1H1, 39HVAR. WITH IDENT. LISTED BELOW HAS MBRS=,I4,
     1          5H,MIN=,I4, 5H,MAX=,I4)
      WRITE(6,511) INPT(9,IJD),(IHEAD(L,IJD),L=1,8)
C
   25 CONTINUE
      REWIND MJD
      IF(NOK.EQ.0) GO TO 710
      CALL JOBPRI(MBUF,MBUF(NX1),MBUF(NX2),NOTLT,MBUF(NX3),MBUF(NX4),
     1 MBUF(NX5), NOK)
  710 CONTINUE
C
C  IF END OF FILE ON CARD READER NOT YET READ, RETURN TO BEGINNING
C
      IF(JDFLAG) 60,8,60
60    CONTINUE
      REWIND NT1
      RETURN
      SUBROUTINE JOBPRI(NWORDS,AVE,ANUMP,NOTLT,IHEAD,NMASK,LTIT,NOK)
      DIMENSION NWORDS(1),AVE(1),ANUMP(1),NOTLT(1),IHEAD(1),NMASK(1),
     1LTIT(11,9),JA(50),KOS(1000),KB(1726),KDUTY(26),LABEL(42),
     2AKOS(1000),KE(1000),OPU(2),ITYP(2),LABL(6,2)
      COMMON JA, KOS, KB, KDUTY, LABEL, AKOS, KE, IS, JSEQ
      EQUIVALENCE (JA(3),NCASES),(JA(12),NTASK),(JA(9),NDUTY),
     1(JA(17),MBRS),(JA(20),KLAB),(JA(24),IDSP),(JA(25),ICOMP),
     2(JA(26),IBIN,IDTP),(JA(27),IBCD,IPTP),(JA(28),MAXP),(JA(29),MAXT),
     3(JA(34),KSW),(OPU(1),IPTR),(OPU(2),ITAP),(JA(1),ID)
      EQUIVALENCE  (JA(19),JLAB)
      INTEGER OPU
      REAL MAXP
      DATA LABL/6*' ','TASK','SEQ','NO',3*'.'/
      DATA ITYP / 'DUTY', 'TASK'/
      DATA JOBSPC /'JOBSPC'/
      IO = 25
      ITAP = 26
      IPTR = 6
      MJD = 20
      MBIT = 2**18
      IBASE = 0
      READ(MJD) NTD, (NWORDS(I),I=1,NTD)
      DO 900 IJD = 1, NOK
```

LINE # TEXT...

```
      READ(MJD)(AVE(I),I=1,NTD),(ANUMP(I),I=1,NTD),KSRT,(IHEAD(I),I=1,8)
     1,NC35,(NMASK(I),I=1,NC35),K,(JA(I),I=1,K)
      AMBRS = MBRS
      IF(IJD.EQ.NOK) REWIND MJD
      IF(IDTP.EQ.1) GO TO 602
      DO 200 K = 1, NTD
      AKOS(K) = AVE(K) / AMBRS
  200 CONTINUE
      N50 = 50
      DO 201 K = 1, 8
      JA(36+K) = IHEAD(K)
  201 CONTINUE
      CALL ERTRAN(9,IDATE,ITIME)
      JA(46) = IDATE
      JA(47) = ITIME
      WRITE(IO) IDSP,NTD,(AKOS(K),K=1,NTD),NC35,(NMASK(K),K=1,NC35),JSEQ
     1,N50,(JA(K),K=1,50),(AVE(K),K=1,NTD),(ANUMP(K),K=1,NTD)
  602 CONTINUE
      DO 400 ITASK = 1, NTD
      KB(ITASK) = ITASK
  400 CONTINUE
C  PRINT DUTY DESCRIPTION ONLY         K1=1, K2=1
C  PRINT TASK DESCRIPTION ONLY         K1=2, K2=2
C  PRINT DUTY AND TASK DESCRIPTIONS  K1=1, K2=2
C
      K1 = 2 - MOD(KSW,2)
      K2 = (KSW + 2) / 2
C
      AMBRS = MBRS
      IF(K1.NE.1) GO TO 402
      K = NTASK +1
      IF(KSRT.NE.1) GO TO 401
      CALL LSORT(NDUTY,AVE(K),KB(K))
      IF(K2.EQ.2) GO TO 403
      GO TO 499
  401 CALL LSORT(NDUTY,ANUMP(K),KB(K))
  402 IF(K2.NE.2) GO TO 499
      IF(KSRT.NE.1) GO TO 404
  403 CALL LSORT(NTASK,AVE,KB)
      GO TO 499
  404 CALL LSORT(NTASK,ANUMP,KB)
  499 CONTINUE
      DO 500 KOP = K1, K2
      LPAGE = 1
      DO 1023 L = 1, IPTP
      IOUT = OPU(L)
      WRITE(IOUT,1010) (IHEAD(K),K=1,8), IDSP, LPAGE
      IF(KLAB.NE.0) WRITE(IOUT,1011) (LABEL(K),K=1,KLAB)
      WRITE(IOUT,1012) ITYP(KOP), NCASES, NTASK, NDUTY, MBRS
      WRITE(IOUT,1017) (LABL(K,KOP),K=1,6)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
1023   CONTINUE
       ASSIGN 407 TO ITYPE
       IF(KOP.EQ.2) ASSIGN 409 TO ITYPE
       KLINE = JLAB + 13
       KEND = NTASK
       KBEG = 1
       GO TO (10,11), KOP
10     KBEG  = NTASK + 1
       KEND  = NTD
11     CONTINUE
       D = 0.0
       JTASK = 0
       DO 450 INDEX = KBEG, KEND
       JTASK = JTASK + 1
       I = KB(INDEX)
       A = ANUMP(I)/AMBRS * 100.0
       IF(A.EQ.0.0) GO TO 500
       B = AVE(I) /ANUMP(I)
       C = AVE(I) /AMBRS
       D = D + C
       LENG = NWORDS(I)
       NL = (LENG-1) / 11 + 1
       IFROM = I + IBASE
       CALL DSKRED(LENG,LTIT,IFROM,1)
       KLINE = KLINE + NL
       IF(KLINE.LE.55) GO TO 405
       KLINE = 2
       LPAGE = LPAGE + 1
405    DO 410 IND = 1, IBCD
       IOUT = OPU(IND)
       IF(IND.EQ.2) GO TO 406
       IF(JTASK.GT.MAXT) GO TO 410
       IF(D.GT.MAXP) GO TO 410
406    IF(KLINE.EQ.2) WRITE(IOUT,1013) IDSP, LPAGE
       GO TO ITYPE, (407,409)
407    WRITE(IOUT,1014) (LTIT(J,1),J=1,11), A, B, C, D
       IF(NL.EQ.1) GO TO 410
408    WRITE(IOUT,1015) ((LTIT(J,K),J=1,11),K=2,NL)
       GO TO 410
409    IF(MOD(JTASK,5).NE.0) GO TO 407
       WRITE(IOUT,1014) (LTIT(J,1),J=1,11), A, B, C, D, JTASK
       IF(NL.GT.1) GO TO 408
410    CONTINUE
450    CONTINUE
500    CONTINUE
703    CONTINUE
       IF(IPTP.EQ.1) GO TO 900
       N = 0
       CALL REPEND(ID,IDSP,IHEAD,JSEQ)
900    CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          RETURN
   1010   FORMAT('1',8A6,29X,A6,' PAGE ',I4,/)
   1011   FORMAT(1X,13A6,A2)
   1012   FORMAT('0',A4,' JOB DESCRIPTION',6X,'CASES    TASKS   DUTIES     M
         1BRS',/,26X,4(I6,3X),/)
   1013   FORMAT('ID-TSK',21X,'DUTY/TASK TITLE',36X,A6,' PAGE ',I3,/)
   1014   FORMAT(1X,11A6,4F7.2,I5)
   1015   FORMAT(1X,11A6)
   1017   FORMAT(27X,  'CUMULATIVE SUM OF AVERAGE PERCENT TIME SPENT BY ALL
         1MEMBERS',07('.'),5X,A4,/,27X,  'AVERAGE PERCENT TIME SPENT BY ALL
         2MEMBERS',18('.'),6X,'.',5X,A3,/,27X,'AVERAGE PERCENT TIME SPENT BY
         3 MEMBERS PERFORMING',4('.'),2(6X,'.'),5X,A2,/,27X,'PERCENT OF MEMB
         4ERS PERFORMING',16('.'),3(6X,'.'),6X,A1,/,71X,4('.',6X)  ,A1,/,
         5' D-TSK',21X,'DUTY/TASK TITLE',29X,4('.',6X)  ,A1,/)
          END
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C
C
          DIMENSION KOS(7000),ZU(7000),MEMBER(7000),YU(7000),
     1          KA(50), KC(7000), KE(1726), ITST(2), KD(1000), AKC(1)
          DIMENSION KT(15,20)
         EQUIVALENCE (AKC,KC)
         DATA KOU9, KPTM / 'COU9 =', ' KPATM'/
            INTEGER  OPT
          CNST = 1000.0
         DATA ITST, JRD /'END OF',' DATA ','KPATM '/
         CALL SETUP(10,11,15,0,0,0,0,0,0,0,0,JSEQ)
         IQR = 6
         IPT = 10
         IBCD = 11
         OPT  = 15
         READ (5,10) KSID, JSID
   10    FORMAT(7X,A6,A2)
         REWIND IPT
         REWIND OPT
         REWIND IBCD
                   READ (IPT) KK,(KA(L),L=1,KK)
         IF(KSID.EQ.KA(1).AND.JSID.EQ.KA(2)) GO TO 30
            WRITE (IQR,20) JRD,KSID,JSID,KA(1),KA(2)
    20      FORMAT ( 1H1,12HID.ERR,CRD= 3A6, 12H,TP.= KPATM ,2A6,
     1              17H)RUN TERMINATED.= )
         STOP
   30    CONTINUE
         MBIT = 2**18
         READ(IBCD) ID1, ID2, KA(6), KX, KA(8)
         IF(ID1.EQ.KA(1).AND.ID2.EQ.KA(2)) GO TO 33
         WRITE(6,32) ID1, ID2, KA(1), KA(2)
   32    FORMAT('ISTAGE DATA TAPE-ID = ',A6,A2,'. HISTORY TAPE-ID = ',
        1A6,A2)
         STOP
   33    NCASES = KA(3)
         NTASK = KA(12)
C  ON NORMAL CODAP RUNS THE LEFT HALF OF KX IS ZERO.  ON PROFILE IT IS
C  NON-ZERO, SPECIFICALLY, THE OVERLAP OPTION. HENCE KA(14) IS A FLAG.
         KA(14) = KX / MBIT
         KA(7) = MOD(KX,MBIT)
          DO 3 KU = 1,NCASES
          MEMBER(KU) = 0
    3     CONTINUE
                   LAST = NCASES-1
            DO  150  KUB  =1,LAST
                   KU = NCASES - KUB
C
C       READ A STAGE RECORD
C
         READ (IBCD) ISTAGE, IBEST, JBEST, IW, JW, BEST, AVE
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          IF(ISTAGE.NE.KU) GO TO 9003
          ZU(ISTAGE) = BEST
          YU(ISTAGE) = AVE
    120   IF(MEMBER(IBEST)/MBIT.EQ.0) GO TO 140
          IBEST = MEMBER(IBEST)/MBIT
          GO TO 120
    140   MEMBER(IBEST) = MEMBER(IBEST) + JBEST*MBIT
          MEMBER(JBEST) = MEMBER(JBEST) + ISTAGE
    150   CONTINUE
          CALL FREE(IBCD)
           ZU(NCASES) = 100.0
           YU(NCASES) = 100.0
           IF (KA(14).EQ.0) GO TO 155
           ZU(NCASES) =0.0
           YU(NCASES) =0.0
     155   CONTINUE
                    KP = 1
                    KOS(1) = 1 +1*MBIT
              DO  160 K =2,NCASES
                    KP = MEMBER(KP)/MBIT
                    KOS(K) = MOD(MEMBER(KP),MBIT) +1 +KP*MBIT
     160   CONTINUE
                   IK= KA(4)
                    WRITE (OPT) KK,(KA(L),L=1,KK)
                    READ (IPT) K,(KC(I),I=1,K),L,(KD(I),I=1,L)
                    WRITE(OPT) K,(KC(I),I=1,K),L,(KD(I),I=1,L)
          K9 = 0
                    DO 300 KJ=1,IK
                    READ(IPT) K,((KT(I,L),I=1,15),L=1,K)
    C   K9 = 1 IS A FLAG WHICH MEANS THIS TAPE WAS MADE AFTER THE 1 NOV. 71
    C     REVISION OF INPSTD AND HAS SPACE ALLOCATED FOR THE KPATH
    C     SEQUENCE NUMBER, C009
          IF((KT(1,9).EQ.K009).AND.(KT(2,9).EQ.KPTH)) K9 = 1
                    WRITE(OPT)K,((KT(I,L),I=1,15),L=1,K)
     300            CONTINUE
    C
          IO = 29
          READ(IPT) NC,(KC(I),I=1,NC),NH,(KD(I),I=1,NH)
          LENGTH = NC + NH + NTASK + 3
          DEFINE FILE IO(NCASES,LENGTH,U,NEXT)
          BACKSPACE IPT
    C
             DO 500  KUB  = 1,NCASES
                DO 400 K = 1,NCASES
                 IF ( KOS(K)/MBIT.EQ.KUB) GO TO 410
     400        CONTINUE
          K = NCASES
     410     READ (IPT) J,(KC(JJ),JJ=1,J),L,(KD(LL),LL=1,L),M,(KE(MM),
         1   MM = 1,M)
                IF (KC(1).NE.(KOS(K)/MBIT))GO TO 9000
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          KC(3) = MOD(KOS(K),MBIT)
        WRITE(10'K) J,(KC(JJ),JJ=1,J),L,(KD(LL),LL=1,L),M,(KE(MM),
     1  MM = 1,M)
  500   CONTINUE
        DO 600  K = 1,NCASES
         READ(10'K) J,(KC(JJ),JJ=1,J),L,(KD(LL),LL=1,L),M,(KE(MM),
     1   MM = 1,M)
        AKC(4) = ZU(K)
        AKC(5) = YU(K)
       IF(K9.EQ.1) AKC(9) = K
         WRITE(OPT) J,(KC(JJ),JJ=1,J),L,(KD(LL),LL=1,L),M,(KE(MM),
     1   MM = 1,M)
  600   CONTINUE
         READ (IPT)  L,(KC(LL),LL=1,L),M,(KD(LL),LL=1,M),N,(KE(LL),
     1            LL= 1,N)
        IF (ITST(1).NE.KC(1).OR. ITST(2).NE.KC(2)) GO TO 9005
         WRITE(OPT)  L,(KC(LL),LL=1,L),M,(KD(LL),LL=1,M),N,(KE(LL),
     2            LL= 1,N)
           KM = 1
        IF (KA(14).NE.0) KM=2
        DO 700 K=1,KM
  610   READ (IPT) L,(KC(LL),LL=1,L)
        WRITE  (OPT) L,(KC(LL),LL=1,L)
        IF (ITST(1).EQ.KC(1).AND.ITST(2).EQ.KC(2)) GO TO 700
        GO TO 610
  700    CONTINUE
        END FILE OPT
        REWIND OPT
        REWIND IPT
        CALL FREE(IPT)
       WRITE(6,9007) LAST
 9007  FORMAT(' *** KPATH COMPLETE ON ',I5,' STAGES')
        STOP
 9000      WRITE (IQR,9002)
       STOP
 9002  FORMAT   ( 49H1KPATH SEQUENCE,TAPE SEQUENCE ERROR,RUN ABORTED. )
 9003     WRITE (IQR,9004) KSID,JSID,KA(6),ISTAGE,KA(1),KA(2),KA(6 ),KU
       STOP
 9004  FORMAT   (24H1COMPARE ERROR STG.CRD=    ,2A6,1X,A6,I5,
     1                                7H,CORE=    ,2A6,1X,A6,I5, 1H )
 9005  WRITE(IQR,9006)
       STOP
 9006  FORMAT (65H1NUMBER OF RECORDS ON TAPE AND STAGE CARDS DIFFER, RUN
     1 ABORTED.) )
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
C      PROGRAM TO PRINT EITHER THE PROFILE OR TASK MATRIX WHICH HAS
C      BEEN OVERLAPPE AND OUTPUT FOR GROUPING.
C
C
C
C
       COMMON/ICOMM/JA,MFLGI,ATAG(100)
C
       DIMENSION JA(50), AKOS(150),  ACARD(84),CNTHCD(14)
        DIMENSION  AKOS2(150)
        DIMENSION  APICK(150)
        DIMENSION  IPICK(150)
        DIMENSION  IPU(2)
        EQUIVALENCE (IPICK,APICK)
        EQUIVALENCE  (APICK(1),AKOS(2))
        EQUIVALENCE  (JA(3),NCASE), (JA(3),NCASC),(JA(1),SID)
        DIMENSION  TITL (8)
        EQUIVALENCE    (CNTHCD(6), TITL(1))
   10   FORMAT (1H1,8A6,29X, A6,6H PAGE  ,2X,I4 //)
    11  FORMAT   (1H ,13A6,A2)
   16   FORMAT (1H ,8(A6))
     20 FORMAT  (13A6,A2)
   13   FORMAT (1H ,A6,12(2X,F8.3))
   14   FORMAT (1H ,A6,12(1X,E9.3))
  15   FORMAT (1H0,6X,12(4X,A6))
        ITAG = 0
        MBIT =  262144
        IPU(1) =6
        IPU(2)=26
        CALL DISK(300,465)
        CALL SETUP(27,26,0,0,0,0,0,0,0,0,0,JSEQ)
        NROW  = 0
        IPAGE = 1
         NVETR = 300
C
C
       READ(5,21) NTEXC,MFLGI,LBCDSW,ISWT,REPOT,TITL
    21 FORMAT(6X,4I2,X,A6,X,8A6)
       NLPAG=55
       IF(LBCDSW.EQ.0) LBCDSW = 2
       IF(ISWT.EQ.0) ISWT = 2
        IF(MFLGI.EQ.0) MFLGI=2
       IF(NTEXC.EQ.0) GO TO 18
        NN = NTEXC*14
        READ (5,20) (ACARD(I),I=1,NN)
    18 CONTINUE
       CALL DISKLD
       CALL OVLAP
       NCA = JA(3)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
         IPGCNT  = NTEXC+5
         NTR  = 1
         MTR  =  12
         IF(NCA .GT. 12) GO TO 111
         MTR = NCA
  111    CONTINUE
         DO 499  LL = 1,LBCDSW
          LK = IPU(LL)
          WRITE  (LK,10) TITL, REPOT,IPAGE
         IF(NTEXC.EQ.0) GO TO 112
          WRITE  (LK,11) (ACARD(I),I = 1,NN)
  112     WRITE  (LK,15) (ATAG(I),I= NTR,MTR   )
  499    CONTINUE
  500     CONTINUE
        CALL DSKRED(NCASE+1,AKOS(1),NVETR,0)
        NVETR = NVETR - 1
  118    MNR = 1
         JIM = (NCASE+1)/2
         DO 119  IIP =1,JIM
           AKOS2(MNR) = IPICK(IIP)/MBIT
          AKOS2(MNR) = AKOS2(MNR)/1000.0
         MNR = MNR+1
           AKOS2(MNR) = MOD(IPICK(IIP),MBIT )
          AKOS2(MNR) = AKOS2(MNR)/1000.0
         MNR = MNR+1
  119    CONTINUE
C  THIS SECTION  WRITE A SECTIONED PORTION OF THE MATRIX IN ONE
C        OF THREE FORMATS,  XX.XXX OR XXXX.XXX  OR .XXXXEXX
  120    ITAG = ITAG+1
         DO 121   LL = 1,LBCDSW
          LK = IPU(LL)
           GO TO (126,127),ISWT
  126    CONTINUE
           WRITE (LK,14) ATAG(ITAG),(AKOS2(JJX),JJX= NTR,MTR)
           GO TO 121
  127    CONTINUE
           WRITE (LK,13) ATAG(ITAG),(AKOS2(JJX),JJX= NTR,MTR)
  121    CONTINUE
  135    NROW =  NROW+1
         IF (NCASE-NROW) 138,138,136
  136    IPGCNT =IPGCNT+1
         IF (NLPAG-IPGCNT) 137,137,500
  137    IPGCNT = NTEXC+5
         IPAGE = IPAGE+1
         GO TO 111
  138    IF (NCASE-MTR) 900,900,140
  140    NTR = NTR+12
         MTR = MTR+12
         IF ((NCA-NTR).LT.12) MTR = NCA
  143    ITAG = 0
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      NROW = 0
      NVETR = 300
      IPAGE = IPAGE+1
      IPGCNT = NTEXC+5
      GO TO 111
  900 IF (LBCDSW.EQ.1) GO TO 901
      CALL REPEND(SID,REPOT,TITL,JSEQ)
  901 WRITE(6,902)
  902 FORMAT('1')
      STOP
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE OVLAP
      INTEGER TIMB
      DIMENSION          TASKB(101),IBUF(930),JBUF(930),ISID(2)
      COMMON/ICOMM/ICOM(50),MFLG1,DUMMY(100)
      EQUIVALENCE  (ICOM(1),ISID(1)),(ICOM(3),NCASE),
     *             (ICOM(10),NTRAK),(ICOM(12),NTASK)
      INTEGER TASKB
      IDTRAK = 300
      MBIT = 262144
      IREAD = 0
      JREAD = 0
      IHOLD = 0
      IB = 1
      ITRAK = 1
      IROW=1
      JROW=1
      JTRAK = 1
      JB=1
 2000 CALL DSKRED( 930 ,IBUF,ITRAK,0 )
      IREAD = 1
 2001 CONTINUE
      IF( IBUF(IB).EQ.IROW) GO TO 20
      PRINT 11
   11 FORMAT( 24HIERROR ROW ID OUT OF SEQ )
      STOP
   20 CONTINUE
      NZI = IBUF(IB+1)
 3000 IF(JREAD.NE.0) GO TO 3001
      CALL DSKRED( 930 ,JBUF,JTRAK,0)
      JREAD = 1
 3001 CONTINUE
      IKOUNT = 0
      IF( JBUF(JB).EQ.JROW) GO TO 30
      WRITE(6,12)
   12 FORMAT(17HIJ ROW OUT OF SEQ )
      WRITE(6,98)JBUF
   98 FORMAT(1X,50Z0)
      WRITE(6,99)IB,JB,IREAD,JREAD,ITRAK,JTRAK,IROW,JROW
   99 FORMAT(1X,4I6)
      STOP
   30 CONTINUE
      NZJ = JBUF(JB+1)
      IF( JROW.NE.IROW) GO TO 100
       TASKB(JROW+1) = 100000
      JJJ = JB +NZJ + 1
      GO TO 200
  100 CONTINUE
      JJ = JB + 2
      II = IB + 2
      JJJ =  JB + NZJ + 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      TIMB=0
      DO 110 J = JJ,JJJ
      JVEC =IABS(JBUF(J)/MBIT)
  210 IVEC=IABS(IBUF(II)/ MBIT)
      IF (JVEC-IVEC ) 110,7,8
    8 II = II + 1
      IF(II.GT.(IB+NZI+1)) GO TO 120
      GO TO 210
    7 IKOUNT = IKOUNT + 1
      IF(MFLG1.EQ.1) GO TO 110
      IPER = MOD(IABS(IBUF(II)),MBIT)
      JPER = MOD(IABS(JBUF(J)),MBIT)
      TIMB=TIMB+MIN0(IPER,JPER)
  110 CONTINUE
  120 CONTINUE
      JROW1=JROW+1
      IF(MFLG1.EQ.1) GO TO 121
      TASKB(JROW1) =TIMB
      GO TO 200
  121 CONTINUE
      TASKB(JROW1)= (IKOUNT * 100000)/NZI + (IKOUNT*100000 )/NZJ
      TASKB(JROW1)= (TASKB(JROW1)+1)/ 2
  200 CONTINUE
      IF( JBUF(JJJ).LT.0) GO TO 202
      JB = JB + NZJ +2
      JROW = JROW + 1
      GO TO 220
  202 JREAD = 0
      JB = 1
      JTRAK = JTRAK +2
      JROW= JROW +1
  220 IF( JROW.LE.NCASE ) GO TO  3000
      TASKB(1)=IROW
      IN = 1
      DO 4000 I=1,NCASE,2
      IN = IN +1
 4000 TASKB(IN) = TASKB(I+2)+ TASKB(I+1)* MBIT
      NCA = NCASE +1
      CALL DSKSTO(NCA , TASKB,IDTRAK,0 )
      IDTRAK = IDTRAK - 1
      JTRAK = 1
      JREAD  = 0
      JB = 1
      JROW = 1
      III = IB+NZI+1
      IF( IBUF(III).LT.0) GO  TO 301
      IB = III + 1
      IROW = IROW + 1
      GO TO 302
  301 IREAD = 0
```

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      IB = 1
      ITRAK = ITRAK + 2
      IF(IROW.EQ.NCASE ) GO TO 1000
      IROW = IROW + 1
  302 CONTINUE
      IF(IREAD.EQ.0 ) GO TO 2000
      GO TO 2001
 1000 NTRAK=1
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE DISKLD
      DIMENSION IBUF(930),IDSP(2026),OUT(100),TT(2),ED(2)
      DIMENSION INBUF(15),AK(2026),NMASK(600)
      EQUIVALENCE (IDSP(1),TREP),(IDSP(2),NTASK),(IDSP(3),AK(1))
      COMMON/ICOMM/JA(50),MFLG1,OUT
      INTEGER CSEQ,TSEQ
      INTEGER OUT
      INTEGER TREP,CREP
      DATA ED/12HEND OF DATA /
      DATA MBIT/262144/,TT/1HT,1HP/,K58/58/,JTRAK/1/
      IBUFKT=0
       JD=27
C
C
   19 ICARD=0
      NCASE = 0
      LASTCD=1
      REWIND JD
C
C     REWIND JOB DESCRIPTION TAPE, READ 1 REQUEST CARD
C
   15 ICARD=ICARD+1
       READ(5,102,END=57) INBUF
  102 FORMAT(A4,9A6,4A4,2X,I4)
      IF(ICARD.LE.100) GO TO 100
      PRINT 103
      STOP
  100 CREP=INBUF(2)
      CSEQ=INBUF(15)
       IF(CSEQ.LT.TSEQ) REWIND JD
C
C     SEARCH JOB DESCRIPTION TAPE FOR REQUESTED REPORT
C
   20 READ(JD,END=85) TREP,NT,(AK(I),I=1,NT),NM58,(NMASK(I),I=1,NM58),
     1TSEQ,IR,(JA(I),I=1,IR)
       NTASK = JA(12)
      IF(TSEQ-CSEQ) 20,25,85
   25 IF(TREP.NE.CREP) GO TO 90
      IF(NTASK.GT.2026) GO TO 95
      NCASE = NCASE + 1
C
C     IF TOTAL NON-ZERO TASKS EXCEED 463, EACH CASE REQUIRES 2 TRACKS
C
   30 CONTINUE
C
C     CONVERT AK ARRAY TO INTEGER PC*1000, FIND NON-ZERO VALUES,
C     AND PACK INTO HALF-WORD FORMAT
C
   35 DO 40 I=1,NTASK
   40 IDSP(I+2)=AK(I)*1000.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      NZE=0
      DO 45 I=1,NTASK
      J=1
      IF(IDSP(I+2).EQ.0) GO TO 45
      NZE=NZE+1
      IDSP(NZE+2)=IDSP(I+2)+J*MBIT
45    IDSP(2)=NZE
      NP2 = NTASK + 2
C
C     FILL UP IBUF ARRAY FOR OUTPUT TO DISK, 2 TRACKS PER ACCESS
C
      OUT(NCASE) = CREP
      LAST=IBUFKT+NZE+2
      IF(LAST.GE.930) GO TO 60
   50 NZE2=NZE+2
      IDSP(1) = NCASE
      DO 55 I=1,NZE2
      IBKT=IBUFKT+I
   55 IBUF(IBKT)=IDSP(I)
      IBUFKT=LAST
      GO TO 15
57    LASTCD = 2
C
C     FLAG LAST WORD NEGATIVE AND OUTPUT 2 TRACKS TO DISK
C
   60 IBUF(IBUFKT)=-IBUF(IBUFKT)
      CALL DSKSTO(930,IBUF,JTRAK,0)
      JTRAK=JTRAK+2
      IBUFKT=0
      LAST=NZE+2
      GO TO (50,75),LASTCD
   75 CONTINUE
       REWIND JD
      JA(3) = NCASE
      RETURN
   85 PRINT 105,   CSEQ,TSEQ
      REWIND JD
      GO TO 15
   90 PRINT 106,   CREP,TREP
      GO TO 15
   95 PRINT 107
      STOP
 103  FORMAT('1NUMBER OF REQUEST CARDS EXCEEDS 100.  RUN ABORTED.')
  105 FORMAT(90H0TAPE SEQUENCE NO. AND CARD TAPE SEQUENCE NUMBER DO NOT
     1MATCH. THIS REQUEST OMITTED. CSEQ=I4, 5HTSEQ=I4)
  106 FORMAT(76H0TAPE REPORT ID AND CARD REPORT ID DO NOT MATCH. THIS RE
     1QUEST OMITTED. CREP=A6, 5HTRAP=A6)
  107 FORMAT(52H0TASKS ON BIN JOB DESC TAPE EXCEEDS MAX. JOB ABORTED)
  110 FORMAT(2A6)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      DOUBLE PRECISION ITIME,HOLD
      DIMENSION KOMM(50),LMNTOP(3501,5),NXTROW(1700),LROWTI(1700,5),
     1 LROWTV(1700,5), KSAT(1700), IMAGE(4), JALT(9)
      DIMENSION MSG(5), LINE(2), LEAD(2), IROW(5), NOTROW(5)
      EQUIVALENCE (LINE(1),NOTCOL),(LINE(2),KSAT(1)),(NOTCOL,IEND),
     1 (LEAD(1),NOTNXT),(LEAD(2),NXTROW(1))
      DATA IBLANK,IZERO,LTIME,LTASK/1H ,1H0,6HTIME  ,6HTASK  /
      DATA NEND1, NEND2 /6HEND OF, 6H DATA /
      DATA KPOINT/0/, LIPLOP/2/, IPRIOR/0/, HOLD/0.0D0/
      DATA MSG/' CODAP   OVRLAP  PREPROCESSING'/
      ASSIGN 2410 TO JUMP
      LENM = 5
      CALL ERII('L',LENM,MSG,ICKOVR)
C
C  ASSIGN I/O AND WORK UNITS.
      CALL SETUPO(KSID,JUMPS)
      INU = 10
      LOP = 15
      LIO = 21
C  REWIND UNITS.
      REWIND INU
      REWIND LIO
C
C
C  READ CONTROL CARD.
      READ(5,805) LSID1,LSID2,MOP,N
      N = 5
C  CHECK MATRIX TYPE CONTROL.
      IF(MOP.EQ.IBLANK .OR. MOP.EQ.IZERO) GO TO 100
C  CREATE TASK MATRIX.
      ASSIGN 1600 TO JMPTYP
      ASSIGN 1900 TO LODTYP
      MID = LTASK
      GO TO 200
100   CONTINUE
C  CREATE TIME MATRIX.
      ASSIGN 1500 TO JMPTYP
      ASSIGN 2000 TO LODTYP
      MID = LTIME
C
200   CONTINUE
C  READ COMMUNICATION REGION (COMM) FROM INPUT TAPE.
      READ(INU) NUM,(KOMM(INDEXA),INDEXA = 1,NUM)
C  CHECK STUDY I.D. .
      IF(KOMM(1).EQ.LSID1 .AND. KOMM(2).EQ.LSID2) GO TO 300
C  STUDY I.D. NO MATCH - WRITE MESSAGE AND ABORT.
      WRITE(6,810) KOMM(1),KOMM(2),LSID1,LSID2
      CALL ERIIX
      CALL ERTRAN(2)
C
```

LINE # TEXT...

```
300   CONTINUE
C  STUDY I.D. MATCH - TAKE INFORMATION FROM COMM.
      NCASE = KOMM(3)
      NDICT = KOMM(4)
      MAXLRL = KOMM(12) + 1
C  CREATE INTERMEDIATE MASS STORAGE SCRATCH UNIT.
      CALL DEFINE(NCASE,MAXLRL,KSID)
      WRITE(6,8181)
8181  FORMAT(1H1)
C  ASSIGN MATRIX TYPE TO COMM.
      KOMM(6) = MID
C  GENERATE NUMBER OF WORDS REQUIRED FOR PACKED OUTPUT MATRIX ROW.
C  NCASE ELEMENTS + ROW NO. REQUIRE  ((NCASE + 1) - 1) / 2  + 1  =
C  NCASE / 2  +1    WORDS
      NCASEH = NCASE/2 + 1
C  COMPUTE NUMBER OF GROUPS OF N OR LESS AND LAST SIZE.
      NCASEB = (NCASE + N - 1)/N
      LASTNB = NCASE - (NCASEB - 1)*N
C  TRANSFER COMM TO OUTPUT DISK.
      WRITE(LIO) NUM,(KOMM(INDEXA),INDEXA = 1,NUM)
C
C  BYPASS RELATIVE LOCATION TABLE AND DICTIONARY ON INPUT TAPE.
      LIMIT1 = NDICT + 1
      DO 600 INDEXB = 1,LIMIT1
      READ(INU) MDUMMY
600   CONTINUE
C
C  FOR EACH CASE - BYPASS COMPUTED AND HISTORY VARIABLES AND WRITE THE
C  NUMBER OF NONZERO TASKS AND TASK DATA ARRAY TO DISK.
      DO 700 INDEXB = 1,NCASE
      READ(INU)  IEND,(KSAT(INDEXA),INDEXA = 1,IEND) , IEND,(KSAT(INDEXA
     1),INDEXA = 1,IEND) ,  IEND,(KSAT(INDEXA),INDEXA = 1,IEND)
      LRL = IEND + 1
      CALL WRITER(INDEXB,LRL,LINE)
700   CONTINUE
C  CHECK END OF DATA - ABORT FOR ERROR.
      READ(INU) J,JEN,JBD
      IF(JEN.EQ.NEND1 .AND. JBD.EQ.NEND2) GO TO 701
      WRITE(6,813) LSID1,LSID2,NCASE
      CALL ERIIX
      CALL ERTRAN(2)
701   CONTINUE
      CALL CLOSER
C
C  THROUGH WITH INPUT - REWIND INPUT UNIT.
      REWIND INU
      CALL FREE(INU)
C
C
C  WRITE "ZERO ROW" LINE TO PRINTER.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      JZERO = 0
      CALL ERTRAN(9,KDATE,KTIME)
      KDATE1 = KDATE
      KTIME1 = KTIME
      DECODE(830,KTIME) LHH,LMM,LSS
      WRITE(6,820) LHH,LMM,LSS,JZERO
      CALL GETPCT(26,IPRIOR)
C  INITIALIZE ROW NUMBER ARRAY.
      DO 750 INDEXA = 1,N
      IROW(INDEXA) = INDEXA
750   CONTINUE
      LAST = N
C
C
C
C
C  OVERLAP PROCESSOR -  1-9 CASES IN TURN ARE HELD WHILE ALL CASES
C  ARE SERIALLY OVERLAPPED WITH THEM TO FORM N MATRIX ROWS.
C
      DO 2500 IROWB = 1,NCASEB
      CALL ERTRAN(9,KDATE,KTIME)
      ENCODE(8107,MSG) KTIME,IROWB,NCASEB
8107  FORMAT(1X,A6,' OVRLAP ',I4,'/',I4,' DONE')
      IF(IROWB.EQ.NCASEB) LAST = LASTNB
      DO 920 INDEXB = 1,LAST
      CALL READR(IROW(INDEXB),NOTNXT,LEAD)
      NOTROW(INDEXB) = NOTNXT
C  SPLIT OFF HALFWORD TASK NO.'S (=I) AND VALUES (=V) TO FULLWORD
C  ARRAYS.  NOTE NXTROW AND LEAD ARE EQUIVALENT.
      DO 900 INDEXA = 1,NOTNXT
      LROWTV(INDEXA,INDEXB) = FLD(18,18,NXTROW(INDEXA))
      LROWTI(INDEXA,INDEXB) = FLD( 0,18,NXTROW(INDEXA))
900   CONTINUE
      FLD( 0,18,LMNTOP(1,INDEXB)) = IROW(INDEXB)
920   CONTINUE
C
C  SET LOAD WORD SHIFT ACTUATOR FOR INITIAL LEFTSHIFT TO PLACE ROW
C  NUMBER IN LEFT HALF OF FIRST OUTPUT WORD FOR THIS ROW.
      IFLIP = 2
C  INITIALIZE OUTPUT ARRAY INDEX.
      INDEXC = 1
C
C
C  PROCESS THIS ROW.
C
      DO 2400 ICOL = 1,NCASE
C  READ NEXT CASE INTO COMPARITOR FRAME (COLUMN, OR COL, POSITION).
      CALL READR(ICOL,NOTCOL,LINE)
      DO 2350 INDEXA = 1,LAST
      IF(ICOL.NE.IROW(INDEXA)) GO TO 1000
```

LINE #   TEXT...

```
      C  FOR ROW = COLUMN, SKIP CASE,...
      C  AND PLUG 100 PER CENT INTO OUTPUT...
            LVALUE = 100000
      C  VIA OUTPUT LOADER.
            GO TO 2100
      C
      1000  CONTINUE
      C
      C  INITIALIZE CASE OVERLAP ACCUMULATORS.
            IKSAT = 0
            INROW = 1
            IACCUM = 0
            NNCOMT = 0
      C
      C
      C  CASE OVERLAPPER - ASSUMES STRICTLY INCREASING TASK I.D.'S
      C
      1300  CONTINUE
      C  INCREMENT COL TASK COUNTER.
            IKSAT = IKSAT + 1
      C  IF ALL COL TASKS EXAMINED, LEAVE OVERLAPPER.
            IF(IKSAT.GT.NOTCOL) GO TO 1800
      C  SPLIT OFF COL TASK I.D. .
            LCOLTI = FLD(0,18,KSAT(IKSAT))
      1400  CONTINUE
      C  IF ROW TASK I.D. GREATER THAN COL TASK I.D., LOOK AT NEXT COL TASK.
            IF(LROWTI(INROW,INDEXA).GT.LCOLTI) GO TO 1300
      C  IF ROW TI (TASK I.D.) EQUALS COL TI, OVERLAP ACCORDING TO
      C  MATRIX TYPE.
            IF(LROWTI(INROW,INDEXA).EQ.LCOLTI) GO TO JMPTYP,(1500,1600)
      C  ROW TI MUST BE LESS THAN COL TI, SO LOOK AT NEXT ROW TASK.
      1401  CONTINUE
      C  INCREMENT ROW TASK COUNTER.
            INROW = INROW + 1
      C  IF ALL ROW TASKS EXAMINED, LEAVE OVERLAPPER.
            IF(INROW.GT.NOTROW(INDEXA)) GO TO 1800
            GO TO 1400
      C
      1500  CONTINUE
      C  TIME MATRIX - SPLIT OFF COL TASK VALUE (TV).
            LCOLTV = FLD(18,18,KSAT(IKSAT))
      C  FIND MIN OF ROW TV AND COL TV.
            IADD = MIN0(LROWTV(INROW,INDEXA),LCOLTV)
      C  SUM MINIMA OVER TASKS.
            IACCUM = IACCUM + IADD
            GO TO 1700
      1600  CONTINUE
      C  TASK MATRIX - INCREMENT NUMBER OF COMMON TASKS.
            NNCOMT = NNCOMT + 1
      C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
1700  CONTINUE
C  PRODUCE NEXT ROW TI AND COL TI.
      GO TO 1401
1800  CONTINUE
C
C  PREPARE FOR OUTPUT LOADING BY MATRIX TYPE.
      GO TO LODTYP,(2000,1900)
C
1900  CONTINUE
C  TASK MATRIX - CALCULATE AVERAGE PROPORTION OF COMMON NONZERO
C  TASKS, AND CONVERT TO PERCENTAGE WITH IMPLIED DECIMAL POINT.
      QROW = NOTROW(INDEXA)
      QCOL = NOTCOL
      COMT = NNCOMT
C  AVERAGE PROPORTION = (1/2)*(COMT/QROW + COMT/QCOL)
C  PERCENTAGE = (100)*(AVERAGE PROPORTION)
C  IMPLIED DECIMAL POINT FORM = (1000)*(PERCENTAGE)
      VALUE = (COMT/QROW + COMT/QCOL) * 50000.
C  ROUND AND CONVERT TO INTEGER LOADING VALUE.
      LVALUE = VALUE + 0.5
      GO TO 2100
C
2000  CONTINUE
C  TIME MATRIX - TRANSFER TO LOADING VALUE.
      LVALUE = IACCUM
C
2100  CONTINUE
C
C
C  HALFWORD OUTPUT LOADER
C
C  SHIFT ACTUATOR
      GO TO (2200,2300),IFLIP
C
2200  CONTINUE
C  ENTER LEFTHALF LOADING VALUE INTO LOADING WORD.
      FLD( 0,18,LMNTOP(INDEXC,INDEXA)) = FLD(18,18,LVALUE)
      GO TO 2350
2300  CONTINUE
C  ENTER RIGHTHALF LOADING VALUE.
      FLD(18,18,LMNTOP(INDEXC,INDEXA)) = FLD(18,18,LVALUE)
2350  CONTINUE
      IF(IFLIP.EQ.2) INDEXC = INDEXC + 1
      IFLIP = 3 - IFLIP
C
2400  CONTINUE
C
      DO 2405 INDEXA = 1,LAST
C  WRITE MATRIX ROW TO DISK OUTPUT (NCASEM IS FROM COMM READ SECTION).
      WRITE(LIO) (LMNTOP(INDEXC,INDEXA),INDEXC = 1,NCASEM)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C  SET NEXT N ROW NUMBERS.
          IROW(INDEXA) = IROW(INDEXA) + N
 2405   CONTINUE
C
C  EXECUTE CYCLIC STATUS PRINTOUT AND ALTERNATE CHECKPOINT.
        CALL EHTRAN(9,KDATE,KTIME)
        DECODE(830,KTIME) LHH,LMM,LSS
 830    FORMAT(3A2)
        DECODE(840,KTIME) LH,LM,LS
 840    FORMAT(3I2)
        LATE = LH*3600 + LM*60 + LS
        GO TO JUMP,(2410,2415)
 2410   CONTINUE
        IRLY = LATE
        MULT = 1
        ASSIGN 2415 TO JUMP
 2415   CONTINUE
        LAPSE = LATE - IRLY
        LAPSE = IABS(LAPSE)
        IF(ICKOVR.NE.0) GO TO 2416
        IF(LAPSE.LT.2400) GO TO 2420
 2416   CONTINUE
        ICKOVR = 0
        IRLY = LATE
        MULT = 1
        CALL GETPCT(26,KURR)
 2417   CONTINUE
        IDEL = KURR - IPRIOR
        IF(IDEL.GE.0) GO TO 2419
        IPRIOR = 0
        GO TO 2417
 2419   CONTINUE
        IPRIOR = KURR
        ITIME = HOLD + IDEL
        HOLD = ITIME
        TIME = (ITIME*200)/10**6
        ENCODE(850,JALT(7)) TIME
 850    FORMAT(F10.3,' CPU SEC')
        JALT(1) = '* CHEC'
        JALT(2) = 'KPOINT'
        KPOINT = KPOINT + 1
        LIPLOP = 3 - LIPLOP
        ENCODE(800,IMAGE) KSID,LIPLOP
 800    FORMAT('@CKPT,P  OV',A4,'CKPT',I1,'. . ')
        CALL EHTRAN(6,IMAGE)
        IF(KURR.LT.IPRIOR) CALL GETPCT(26,IPRIOR)
        ENCODE(860,JALT(3)) KPOINT,KSID,LIPLOP
 860    FORMAT(I4,2X,'OV',A4,'CKPT',I1,'.')
        GO TO 2440
 2420   CONTINUE
```

LINE # TEXT...

```
      IF(LAPSE.LT.(1200 * MULT)) GO TO 2460
      MULT = MULT + 1
      DO 2430 INDEXI = 1,9
      JALT(INDEXI) = ' '
2430  CONTINUE
2440  CONTINUE
      WRITE(6,820) LHH,LMM,LSS,IROW(INDEXA),JALT
820   FORMAT(4X,A2,1H:,A2,1H:,A2,6X,'OVRLAP THROUGH THE ',I4,'-TH MATRIX
     1 ROW.',6X,9A6)
2460  CONTINUE
C
2500  CONTINUE
C
C  INITIATE DISK-TO-TAPE TRANSFER.
      ENDFILE LIO
      REWIND LIO
C  WRITE *NCASE ROW* AND NET CPU SEC. LINES AND PUNCH STATISTICS CARD.
      CALL ERTRAN(9,KDATE,KTIME)
      DECODE(830,KTIME) LHH,LMM,LSS
      WRITE(6,820) LHH,LMM,LSS,NCASE
      CALL GETPCT(26,KURR)
2510  CONTINUE
      IDEL = KURR - IPRIOR
      IF(IDEL.GE.0) GO TO 2520
      IPRIOR = 0
      GO TO 2510
2520  CONTINUE
      IPRIOR = KURR
      ITIME = HOLD + IDEL
      HOLD = ITIME
      TIME = (ITIME*200)/10**6
      WRITE(6,870) TIME
870   FORMAT('0',F11.3,' = TOTAL CPU SECONDS TO COMPUTE OVERLAP',
     1 ' MATRIX.')
      WRITE(1,880) KSID,LSID1,LSID2,HID,NCASE,TIME,KDATE1,KTIME1,
     1 KDATE,KTIME
880   FORMAT('OVRLAP ',A4,1X,A6,A2,1X,A6,1X,I6,1X,F11.3,2(2X,A6,1X,A6))
      CALL SETOVR
      GO TO (2550,2700),JUMPS
2550  CONTINUE
      REWIND LOP
      READ(LIO) NUM,(KOMM(INDEXA),INDEXA = 1,NUM)
      WRITE(LOP) NUM,(KOMM(INDEXA),INDEXA = 1,NUM)
      DO 2600 IROWB = 1,NCASE
      READ(LIO) (LMNTOP(INDEXC,1),INDEXC = 1,NCASEM)
      WRITE(LOP) (LMNTOP(INDEXC,1),INDEXC = 1,NCASEM)
2600  CONTINUE
C
C  CLOSEOUT
      ENDFILE LOP
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      REWIND LOP
 2730 CONTINUE
      CALL SETEND
      WRITE(6,815) LSID1,LSID2,NCASE,NCASE,MID
      CALL ERTIX
      STOP
C
C
C
 805  FORMAT(7X,A6,A2,1X,A1,2X,I1)
 810  FORMAT(1H1,4X,'HISTORY TAPE STUDY I.D. DOES NOT MATCH CONTROL CARD
     1 STUDY I.D. .'/5X,'HISTORY TAPE STUDY I.D. IS ',A6,A2,' AND CONTRO
     2L CARD STUDY I.D. IS ',A6,A2,' .'/5X,'RUN ABORTS.')
 813  FORMAT(1H1,4X,'END-OF-DATA DISCREPANCY IN STUDY ',A6,A2,' FOR NCAS
     1E = ',I4,' .'/5X,'RUN ABORTS.')
 815  FORMAT(1H1,4X,'OVERLAP COMPLETE FOR STUDY ',A6,A2,' .'/5X,'OUTPUT
     1FILE CONTAINS A ',I4,' BY ',I4,2X,A6,'MATRIX.'/5X,'OVRLAP ENDS.')
C
      SUBROUTINE SETUPO(ISTUDY,JUMPS)
C  THIS IS A SPECIAL SUBROUTINE FOR THE OVRLAP PROGRAM.
C  IT IS DESIGNED TO ASSIGN OUTPUT TAPES ONLY AFTER ALL
C  CALCULATIONS ARE COMPLETED AND RECORDED ON MASS STORAGE.
C  A BLANK OUTPUT FIELD WILL CAUSE ACCT. TO BE USED FOR
C  FILENAMES AND ACCT-21 TO BE LEFT CATALOGUED, WITH NO TAPE
C  REQUEST.
      DIMENSION NAM1(3), NAM2(3), IMAGE(8), ID(2), ICON(3)
      DATA LD,LI/'D','I'/
      DATA ID/'ID     '/
       LIPLOP = 1
      JUMPS = 1
      IUSE = 10
      ISC2 = 21
      IOUT = 15
      READ(5,100) IDPROG, NAM1, NAM2
      WRITE(6,101) IDPROG
      CALL ASGA(NAM1,IUSE)
      IF(NAM2(1).NE.' ') GO TO 998
      JUMPS = 2
      CALL GETPCT(23,NAM2(1))
 998  ENCODE(102,ISTUDY) NAM2(1)
      IDRF = 'RFILE-'
      ENCODE(105,IMAGE) IDRF,ISTUDY
      CALL ERTRAN(6,IMAGE)
      ENCODE(113,IMAGE) IDRF,ISTUDY
      CALL ERTRAN(6,IMAGE)
      IDRF = 'INDEX-'
      ENCODE(105,IMAGE) IDRF,ISTUDY
      CALL ERTRAN(6,IMAGE)
      ENCODE(113,IMAGE) IDRF,ISTUDY
      CALL ERTRAN(6,IMAGE)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          ASSIGN 7 TO IPICK
          ASSIGN 3 TO JPICK
  999     ENCODE(120,ICON) ISTUDY,LIPLOP
  1       ENCODE(104,IMAGE) ICON
          CALL ERTRAN(6,IMAGE)
          ENCODE(108,IMAGE) ICON
          CALL ERTRAN(6,IMAGE)
          CALL DKSTAT(ID,IC1,IC2,IFLAG)
          ENCODE(112,IMAGE) ICON
          CALL ERTRAN(6,IMAGE)
          IF(IFLAG.EQ.1) GO TO IPICK,(7,11)
          GO TO JPICK,(3,5)
  3       IIF1 = LD
          ENCODE(122,IMAGE) ICON
          WRITE(6,106) (IMAGE(I),I=1,4)
          CALL GETMAS(IMAGE)
          IF(LIPLOP.EQ.1) GO TO 4
          GO TO 9
  4       LIPLOP = 2
          GO TO 999
  5       IIF2 = LD
          ENCODE(123,IMAGE) ICON
          WRITE(6,106) (IMAGE(I),I=1,4)
          CALL GETMAS(IMAGE)
          GO TO 13
  7       ENCODE(103,IMAGE) ISTUDY,LIPLOP
          CALL GETMAS(IMAGE)
          IIF1 = LD
          ENCODE(112,IMAGE) ICON
          CALL ERTRAN(6,IMAGE)
          ENCODE(122,IMAGE) ICON
          WRITE(6,106) (IMAGE(I),I=1,4)
          CALL GETMAS(IMAGE)
          IF(LIPLOP.EQ.1) GO TO 4
  9       ASSIGN 11 TO IPICK
          ASSIGN 5 TO JPICK
          ENCODE(124,ICON) ISTUDY,ISC2
          GO TO 1
  11      ENCODE(109,IMAGE) ISTUDY, ISC2
          CALL GETMAS(IMAGE)
          IIF2 = LD
          ENCODE(112,IMAGE) ICON
          CALL ERTRAN(6,IMAGE)
          ENCODE(122,IMAGE) ICON
          WRITE(6,106) (IMAGE(I),I = 1,4)
          CALL GETMAS(IMAGE)
  13      ENCODE(111,IMAGE) ISC2, ISTUDY, ISC2
          WRITE(6,106) (IMAGE(I),I=1,3)
          CALL ERTRAN(6,IMAGE)
          RETURN
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
      C  SETOVR
             ENTRY SETOVR
             GO TO (15,17),JUMPS
      15     WRITE(6,107)
             CALL ASGC(NAM2,IOUT)
             RETURN
      17     ENCODE(I12,IMAGE) ICON
             CALL ERTRAN(6,IMAGE)
             RETURN
      C  SETEND
             ENTRY SETEND
             ENCODE(I10,IMAGE) IIF2,ISTUDY,ISC2
             WRITE(6,106) (IMAGE(I),I=1,4)
             CALL ERTRAN(6,IMAGE)
             DO 19 LIPLOP = 1,2
             ENCODE(I15,IMAGE) IIF1,ISTUDY,LIPLOP
             WRITE(6,106) (IMAGE(I),I=1,4)
      19     CALL ERTRAN(6,IMAGE)
             ENCODE(I13,IMAGE) IDRF,ISTUDY
             CALL ERTRAN(6,IMAGE)
             IDRF = 'RFILE-'
             ENCODE(I13,IMAGE) IDRF,ISTUDY
             CALL ERTRAN(6,IMAGE)
             RETURN
      C
      100    FORMAT(A6,1X,2(A6,A4,A1,1X))
      101    FORMAT('1DYNAMICALLY ISSUED CONTROL STATEMENTS FOR ',A6,//)
      102    FORMAT(A4)
      103    FORMAT('@ASG,CP OV',A4,'CKPT',I1,'.,F/0/POS/25 . ')
      104    FORMAT('@ASG,Y',4X,3A6)
      105    FORMAT('@ASG,Y ',A6,A4,'. . ')
      106    FORMAT(1X,8A6)
      107    FORMAT('1OVRLAP COPYING TO OUTPUT TAPE',//)
      108    FORMAT('@USE  ID.,',3A6)
      109    FORMAT('@ASG,U    ',A4,'-',I2,'.,F/0/POS/64 . ')
      110    FORMAT('@FREE,',A1,2X,A4,'-',I2,'. . ')
      111    FORMAT('@USE ',I2,',',A4,'-',I2,'. . ')
      112    FORMAT('@FREE',5X,3A6)
      113    FORMAT('@FREE,D ',A6,A4,'. . ')
      115    FORMAT('@FREE,',A1,'   OV',A4,'CKPT',I1,'. . ')
      120    FORMAT('OV',A4,'CKPT',I1,'. . ')
      122    FORMAT('@ASG,A',4X,3A6)
      123    FORMAT('@ASG,A  ',2X,3A6)
      124    FORMAT(A4,'-',I2,'. . ',7X)
             END
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
 .            WORKING-STORAGE
 .
 .
$(0)          AXR$                              .
ARG           FORM       12,6,18                . CONTROL WORD FORMAT
ASGI                     '@ASG,CR  INDEX-XXXX.,F60///20    .  ' .
ASGR                     '@ASG,CR  RFILE-XXXX.,F14///X     .  ' .
EMSG                     'RFILE ERROR:'         .
                         '
ERR           +          0                      . FAC STATUS OF CSF$ REQUEST
E1            ARG        20,0,M1                . I/O ERROR
              +          0                      .
              ARG        27,4,M1+3              .
              +          0                      .
E2            ARG        15,0,M2                . RECORD NUMBER OUT OF RANGE
              +          0                      .
              ARG        8,5,M2+2               .
              ARG        0,0,RECN               .
              ARG        13,3,M2+4              .
              ARG        0,0,MAXREC+1           .
              ARG        1,0,M2+7               .
              +          0                      . LRL OUT OF RANGE ON WRITE
E3            ARG        33,0,M3                .
              ARG        0,0,LRL                .
              ARG        1,5,M3+5               .
              +          0                      .
E4            ARG        35,0,M4                . ATTEMPT TO READ NON-EXISTANT RECORD
              ARG        0,0,RECN               .
              ARG        10,1,M4+6              .
              +          0                      .
FREE                     '@FREE        INDEX-XXXX. . ' .
HOLD          RES        3                      . TEMPORARY STORAGE FOR "ERRXIT"
IFLAG         +          0                      . 0=INDEX BLOCK UNCHANGED, NON-0=CHANGED
IREC          RES        224                    . CURRENT INDEX BLOCK
JPL           +          '          '           . JOHNNY'S PRINT LINE
              RES        6                      .
LRL           +          0                      . CURRENT LRL
MAXLRL        +          0                      . VALID LRL RANGE (0-N)
              +          0                      .
MAXREC        +          0                      . VALID RECORD NUMBER RANGE (0-N)
              +          0                      .
M1                       '  I/O ERROR XX FROM XX OPERATION ON FILE "RFILE".' .
M2                       '  LAST CALL TO XX RECORD XX.  (MAXREC = XX) '.
M3                       '  INVALID WRITE ATTEMPT.  (LRL = XX)' .
M4                       '  INVALID READ ATTEMPT.  NO RECORD XX ON RFILE.' .
NINDX         +          0                      . NUMBER OF INDEX BLOCKS
NSEC          +          0                      . NUMBER OF SECTORS PER RECORD
NTRAK         +          0                      . NUMBER OF TRACKS FOR RFILE
READI         I$OD       'IN   ',R$  224,IREC  0 . READ INDEX FILE
              RES        2                      .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
READRF     I$OD      'RF    ',R$  000,0000  0 . READ RANDOM FILE
           RES       2                      .
RECN       +         0                      . CURRENT RECORD NUMBER
SVX11      RES       1                      . REGISTER SAVE AREA
SVA0       RES       2                      . SAVE REGISTER A0,A1
SVA2       RES       2                      . SAVE REGISTER A2,A3
SVA4       RES       1                      . SAVE REGISTER A4
USEI                 '@USE     IN.,INDEX-XXXX. . ' .
USER                 '@USE     RF.,RFILE-XXXX. . ' .
WRITI      I$OD      'IN    ',W$  224,IREC  0 . WRITE INDEX FILE
           RES       2                      .
WRITRF     I$OD      'RF    ',W$  000,0000  0 . WRITE RANDOM FILE
           RES       2                      .
$(1)       END$                             .
/.
 .
 .         CALL  DEFINE (MAXREC,MAXLRL,ISTUDY)
 .
 .         LMJ       X11,DEFINE
 .         +         MAXREC           0 MAXIMUM NUMBER OF RECORDS ON FILE
 .         +         MAXLRL           1 MAXIMUM RECORD LENGTH
 .         +         ISTUDY           2 STUDY NUMBER (A4 FORMAT)
 .                   (WB)             3 WALK-BACK WORD
 .                   RETURN           4
 .
DEFINE*    .
           SX        X11,SVX11              .
           DS        A0,SVA0                . SAVE A0,A1
           SA        A4,SVA4                . SAVE A4
           LA        A0,*1,X11              .
           SA        A0,MAXLRL+1            . UPPER LIMIT ON RANGE CHECK
           AA,U      A0,27                  .
           DSL       A0,36                  .
           DI,U      A0,28                  .
           SA        A0,NSEC                . NUMBER OF SECTORS FOR LARGEST RECORD
           DSL       A0,36                  . MOVE NUMBER OF SECTORS TO A1, 0 TO A0
           DI,U      A0,4                   . DIVIDE (A0,A1) BY 4
           JZ        A1,$+5                 . IF NO REMAINDER MAX RECD = MULT OF 4
           LA,U      A0,4                   .  ELSE FORCE MAX RECD TO MULT OF 4
           ANA       A0,A1                  .
           AA        A0,NSEC                .
           SA        A0,NSEC                .
           LA        A0,*0,X11              .
           SA        A0,MAXREC+1            . UPPER LIMIT ON RANGE CHECK
           MSI       A0,NSEC                .
           AA,U      A0,63                  .
           DSL       A0,36                  .
           DI,U      A0,64                  .
           SA        A0,NTRAK               . NUMBER OF TRACKS FOR "RFILE"
           LA        A0,MAXREC+1            .
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
                AA,U      AO,223              .
                DSL       AO,36               .
                DI,U      AO,224              .
                SA        AO,NINDX            . NUMBER OF INDEX BLOCKS
                LA        AO,2,X11            . RETREIVE STUDY NUMBER
                SA,H2     AO,$+3              . STORE ADDR. OF STUDY
                SA,H2     AO,$+5              . STORE ADDR. OF STUDY
                LMJ       X11,MOVFD           . MOVE STUDY # TO @ASG CARD
                ARG       4,0,00              .
                ARG       0,3,ASGI+2          . STORE IN INDEX NAME
                LMJ       X11,MOVFD           . MOVE STUDY # TO @ASG CARD
                ARG       4,0,00              .
                ARG       0,3,ASGR+2          . STORE IN RFILE NAME
                LMJ       X11,MOVFD
                ARG       4,3,ASGI+2          .
                ARG       0,1,USEI+3          .
                LMJ       X11,MOVFD
                ARG       4,3,ASGR+2          .
                ARG       0,1,USER+3          .
                LMJ       X11,MOVBIN          . STORE NUMBER OF TRACKS IN @ASG CARD
                ARG       28,0,ASGR           .
                ARG       0,0,NTRAK           .
                LA,U      AO,USEI             .
                ER        CSF$                . EQUATE IN $ INDEX
                LA,U      AO,USER             .
                ER        CSF$                . EQUATE RF $ RFILE
                LMJ       X11,MOVFD           . MOVE @USE TO PRINT LINE
                ARG       36,0,USEI           .
                ARG       5,1,JPL             .
                LA        AO,(ARG 1,4,JPL)    . PRINT IT, SINGLE SPACED
                ER        PRINT$              .
                LMJ       X11,MOVFD           . MOVE @USE TO PRINT LINE
                ARG       36,0,USER           .
                ARG       5,1,JPL             .
                LA        AO,(ARG 1,4,JPL)    . PRINT IT, SINGLE SPACED
                ER        PRINT$              .
                LMJ       X11,MOVFD           . MOVE @ASG CARD TO PRINT AREA
                ARG       36,0,ASGI           .
                ARG       5,1,JPL             .
                LA        AO,(ARG 1,6,JPL)    . PRINT IT, SINGLE SPACED
                ER        PRINT$              .
                SZ        A4                  . A4 = CSF$ ERROR TEST
        DO1     LA,U      AO,ASGI             .
                ER        CSF$                . ATTEMPT TO ASSIGN "INDEX"
                SA        AO,ERR              .
                JP        AO,DO2              . JUMP IF NOT FAC REJECT
                TEP       AO,(1,0)            .
                J         DO1                 . JUMP BACK ON FACILITIES WAIT
                LMJ       X11,FACREJ          . DISPLAY FAC REJECT MESSAGES
                +         ERR                 .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
                NOP                             .
                LA,U        A4,1                . SET ERROR FLAG
        D02     LMJ         X11,MOVFD           . MOVE @ASG CARD TO PRINT AREA
                ARG         36,0,ASGR           .
                ARG         5,1,JPL             .
                LA          A0,(ARG 1,6,JPL)    . PRINT IT, SINGLE SPACED
                ER          PRINT$              .
        D03     LA,U        A0,ASGR             .
                ER          CSF$                . ATTEMPT TO ASSIGN "RFILE"
                SA          A0,ERR              .
                JP          A0,D04              . JUMP IF NOT FAC REJECT
                TEP         A0,(1,0)            .
                J           D03                 . JUMP BACK ON FACILITIES WAIT
                LMJ         X11,FACREJ          . DISPLAY FAC REJECT MESSAGES
                +           ERR                 .
                NOP                             .
                LA,U        A4,2                . SET ERROR FLAG
        D04     JZ          A4,$+2              .
                ER          ERR$                . ERROR HALT
                LX          X11,NINDX           .
                ANX,U       X11,1               . ADJUST INDEX FOR LOOPING
        D05     I$OW        WRITI               . WRITE INDEX BLOCK OF ZEROS
                LA,H2       A0,WRITI+5          .
                AA,U        A0,8                . UPDATE SECTOR ADDRESS
                SA,H2       A0,WRITI+5          .
                JGD         X11,D05             .
                SZ,H2       WRITI+5             . RESET SECTOR TO ZERO
                LX          X11,SVX11           .
                DL          A0,SVA0             . RESTORE A0,A1
                LA          A4,SVA4             . RESTORE A4
                J           4,X11               . RETURN
        /.      CALL    CLOSER
         .
         .      LMJ         X11,CLOSER
         .      +           (WB)                0 WALKBACK WORD
         .                  RETURN              1
        CLOSER*    .
                SA          A0,SVA0             .
                SX          X11,SVX11           .
                TNZ         IFLAG               .
                J           C01                 . LAST BLOCK NOT CHANGED
                I$OW        WRITI               . WRITE LAST INDEX BLOCK
                TNZ,S1      WRITI+3             .
                J           $+4                 . JUMP ON NORMAL RETURN
                LR          R1,(ARG 5,0,('WRITE ')) .
                LR,S1       R2,WRITI+3          .
                J           INDXERR
        C01     SZ          IFLAG               . SET CHANGE FLAG TO ZERO
                LMJ         X11,MOVFD           .
                ARG         4,3,ASGI+2          .
```

```
LINE #   TEXT...

                 ARG        0,0,FREE+3          .
                 LA,U       AO,FREE             .
                 ER         CSF$                . FREE THE INDEX FILE
                 LA         AO,('RFILE-')
                 SA         AO,FREE+2           .
                 LA,U       AO,FREE             .
                 ER         CSF$                . FREE THE RFILE
                 LA         AO,('@ASG,A')
                 SA         AO,FREE             .
                 LA,U       AO,FREE             .
                 ER         CSF$                . ASG,A THE RFILE
                 LA         AO,('INDEX-')       .
                 SA         AO,FREE+2           .
                 LA,U       AO,FREE             .
                 ER         CSF$                . ASG,A THE INDEX FILE
                 LA,U       AO,USE1             .
                 ER         CSF$                . RE-ESTABLISH THE INTERAL FILE NAMES
                 LA,U       AO,USER             .
                 ER         CSF$                .
                 LA         AO,SVAO             .
                 LX         X11,SVX11           .
                 J          1,X11               . RETURN
        /.       CALL    READR (RECN,LRL,ARRAY)
        .
        .        LMJ        X11,READR
        .        +          RECN                0 RECORD NUMBER
        .        +          LRL                 1 RECORD LENGTH SET BY "READR"
        .        +          ARRAY               2 ARRAY TO RECEIVE RECORD
        .                   (WB)                3 WALK-BACK WORD
        .                   RETURN              4
        READR.   .
                 DS         AO,SVAO             . SAVE AO,A1
                 DS         A2,SVA2             . SAVE A2,A3
                 LA         A2,*0,X11           .
                 DL         AO,MAXREC           .
                 TW         AO,A2               .
                 J          BADR                . RECORD NUMBER IS OUT OF RANGE
                 ANA,U      A2,1                .
                 DSL        A2,36               .
                 DI,U       A2,224              . A2 = INDEX BLOCK NUMBER
                 MSI,U      A2,8                . A2 = SECTOR ADDRESS, A3 = SUBSCRIPT
                 TNE,R2     A2,READI+5          .
                 J          RO2                 . CORRECT BLOCK IS IN CORE
                 SA,R2      A2,READI+5          .
                 I$OW       READI               . READ NEW INDEX BLOCK
                 TNZ,S1     READI+3             .
                 J          $+4                 . JUMP ON NORMAL RETURN
                 LR         R1,(ARG 4,0,('READ  ')) .
                 LR,S1      R2,READI+3          .
                 J          INDXERR             .
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
                SZ         IFLAG                 . NO CHANGES YET
        R02     LA,H2      A0,IREC,A3            . A0 = LRL
                JZ         A0,BADREC             . JUMP IF RECORD NOT PRESENT
                SA         A0,*1,X11             . SET USER'S VARIABLE
                SA,H1      A0,READRF+4           . SET WORD COUNT IN I/O PACKET
                LA         A0,2,X11              .
                SA,H2      A0,READRF+4           . SET STORAGE ADDRESS IN I/O PACKET
                LA,H1      A2,IREC,A3            . PICK UP SECTOR ADDRESS
                SA         A2,READRF+5           . SET SECTOR ADDRESS IN I/O PACKET
                I$OW       READRF                . READ RECORD INTO USER'S ARRAY
                DL         A0,SVA0               . RESTORE A0,A1
                DL         A2,SVA2               . RESTORE A2,A3
                TNZ,S1     READRF+3              .
                J          4,X11                 . NORMAL RETURN
                LR         R1,(ARG 4,0,('READ  ')) .
                LR,S1      R2,READRF+3           .
                J          IOERR                 . ERROR HALT
        /.      CALL    WRITER (RECN,LRL,ARRAY)
        .
        .       LMJ        X11,WRITER
        .       +          RECN                  0 RECORD NUMBER
        .       +          LRL                   1 RECORD LENGTH
        .       +          ARRAY                 2 RECORD ADDRESS
        .                  (WB)                  3 WALK-BACK WORD
        .                  RETURN                4
        .
        WRITER* .
                DS         A0,SVA0               . SAVE A0,A1
                DS         A2,SVA2               . SAVE A2,A3
                LA         A2,*0,X11             .
                SA         A2,RECN               . SAVE RECORD NUMBER
                DL         A0,MAXREC             .
                TW         A0,A2                 .
                J          BADW                  . RECORD NUMBER IS OUT OF RANGE
                ANA,U      A2,1                  .
                DSL        A2,36                 .
                DI,U       A2,224                .
                MSI,U      A2,8                  . A2 = SECTOR ADDRESS, A3 = SUBSCRIPT
                TNE,H2     A2,WRITI+5            .
                J          W02                   . CORRECT INDEX BLOCK IS IN CORE
                TNZ        IFLAG                 .
                J          W01                   . NO CHANGE IN CURRENT BLOCK
                I$OW       WRITI                 . WRITE INDEX BLOCK
                TNZ,S1     WRITI+3               .
                J          $+4                   . JUMP ON NORMAL RETURN
                LR         R1,(ARG 5,0,('WRITE ')) .
                LR,S1      R2,WRITI+3            .
                J          INDXERR               .
        W01     SA,H2      A2,WRITI+5            . SET NEW SECTOR ADDRESS
                SA,H2      A2,READI+5            .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
             I$OW      READI              . READ NEW INDEX BLOCK
             TNZ,S1    READI+3            .
             J         W02                .
             LR        R1,(ARG 4,0,('READ  ')) .
             LR,S1     R2,READI+3         .
             J         INDXERR            .
      W02    LA        A2,*1,X11          .
             SA        A2,LRL             . SAVE LRL
             DL        A0,MAXLRL          .
             TW        A0,A2              .
             J         BADLRL             . INVALID LRL PARAMETER
             SA,H2     A2,IREC,A3         . SET INDEX VALUE IN H2 BLOCK
             SA        A2,IFLAG           . MARK CHANGE TO CURRENT BLOCK
             SA,H1     A2,WRITRF+4        . SET WORD COUNT IN I/O PACKET
             LA        A0,2,X11           .
             SA,H2     A0,WRITRF+4        . SET STORAGE ADDRESS IN I/O PACKET
             LA        A2,RECN            .
             ANA,U     A2,1               .
             MSI       A2,NSEC            .
             SA        A2,WRITRF+5        . SET SECTOR ADDRESS IN I/O PACKET
             SA,H1     A2,IREC,A3         . STORE SECTOR ADDR. IN INDEX
             I$OW      WRITRF             . WRITE RECORD ON "RFILE"
             DL        A0,SVA0            . SAVE A0,A1
             DL        A2,SVA2            . SAVE A2,A3
             TNZ,S1    WRITRF+3           .
             J         4,X11              . NORMAL RETURN
             LR        R1,(ARG 5,0,('WRITE ')) .
             LR,S1     R2,WRITRF+3        .
             J         IOERR              .
      /.     ERROR ROUTINES
      .
      .
      INDXERR .                           . I/O ERROR ON INDEX FILE
             LA        A4,('INDEX"')      .
             SA        A4,M1+7            .
      IOERR  .                            . I/O ERROR -- COMMON CODE
             SR        R1,E1+1            .
             SX        X11,SVX11          .
             LMJ       X11,BINOCT         . STORE ERROR CODE IN MESSAGE LINE
             ARG       13,0,M1            .
             ARG       2,0,R2             .
             LX,U      X9,E1              .
             J         ERRXIT+1           .
      BADR   .                            . BAD RECORD NUMBER ON READ
             LR        R1,(ARG 4,0,('READ  ')) .
             J         $+2                .
      BADW   .                            . BAD RECORD NUMBER ON WRITE
             LR        R1,(ARG 5,0,('WRITE ')) .
             SR        R1,E2+1            .
             LX,U      X9,E2              .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
              J          ERRXIT               .
     BADLRL   .                               . LRL OUT OF RANGE ON WRITE
              LX,U       X9,E3                .
              J          ERRXIT               .
     BADREC   .                               . NO RECORD FOUND ON READ
              LX,U       X9,E4                .
     ERRXIT   .                               . COMMON ERROR HALT
              SX         X11,SVX11            .
              LXI,U      X9,1                 .
              LX         X8,(1,EMSG+2)        . X8 = STORAGE INDEX (WORD ADDRESS)
              LA         A3,(1,0)             . A3 = STORAGE INDEX (BYTE ADDRESS)
     EM01     LA         A4,0,*X9             .
              SA         A4,HOLD              . SAVE CURRENT INSTRUCTION
              JZ         A4,EM04              . END OF MESSAGE
              TZ,T1      HOLD                 .
              J          EM02                 . FD CHARACTER STRING
              LA         A4,*HOLD             .
              SA         A4,HOLD              . HOLD = VALUE TO BE CONVERTED
              LMJ        X11,MOVBIN           . CONVERT TO FD
              ARG        1,0,HOLD             .
              ARG        0,0,HOLD             .
              LR,T1      R1,$-1               . R1 = NC
              LX         X7,(1,HOLD)          . X7 = SOURCE WORD ADDRESS
              LA         A2,(1,0)             . A2 = SOURCE BYTE ADDRESS
              J          EM03                 .
     EM02     LXM        X7,A4                .
              LXI,U      X7,1                 . X7 = SOURCE WORD ADDRESS
              LXM,S3     A2,HOLD              .
              LXI,U      A2,1                 . A2 = SOURCE BYTE ADDRESS
              LR,T1      R1,HOLD              . R1 = NC
              J          EM03                 .
              EX         GETC,*A2             . MOVE CHARACTER STRING
              EX         PUTC,*A3             .
     EM03     JGD        R1,$-2               .
              J          EM01                 .
     EM04     LA         A0,(ARG 4,11,EMSG).    PRINT ERROR MESSAGE
              ER         PRINT$               .
              LX         X11,SVX11            . RESTORE LINKAGE REGISTER
              ER         ERR$                 . HALT
     /.       LITERALS
      .
              END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C     THIS PROGRAM PRINTS THE VARIABLE DICTIONARY FROM THE KPATH OR
C     HISTORY TAPE.  SUBROUTINE PRTDIC IS CALLED ONCE TO PRINT AND
C     CALLED AGAIN TO WRITE ON THE BCD TAPE.
      DIMENSION ILOC(400),ICOMM(50)
       IUNIT=10
       IREPT=26
      IPRNT=6
       CALL SETUP(10,26,0,0,0,0,0,0,0,0,0,JSEQ)
C
C     READ CONTROL CARD
      READ 5, ID1,ID2,NTITLE,IBCD
    5 FORMAT(7X,A6,A2,2I2)
      IF(NTITLE.LE.8) GO TO 100
      PRINT 10
   10  FORMAT('1MORE THAN 8 PAGE HEADING CARDS SPECIFIED.  RUN ABORTED')
      STOP
  100 READ(IUNIT) K,(ICOMM(I),I=1,K)
      IF(ID1.EQ.ICOMM(1).AND.ID2.EQ.ICOMM(2)) GO TO 110
      PRINT 20
   20 FORMAT('1STUDY ID NO MATCH.  RUN ABORTED.')
      STOP
  110 NDICT=ICOMM(4)
      READ(IUNIT)
       IF(NTITLE.EQ.0) GO TO 130
      NWORDS=NTITLE*14
      READ 30,(ILOC(I),I=1,NWORDS)
   30 FORMAT(13A6,A2)
  130 CALL PRTDIC(IUNIT,IPRNT,NTITLE,ILOC,NDICT)
      IF(IBCD.NE.0) GO TO 140
      REWIND IUNIT
      READ(IUNIT)
      READ(IUNIT)
      CALL PRTDIC(IUNIT,IREPT,NTITLE,ILOC,NDICT)
      CALL REPEND(ID1,'DICT  ','VARIABLE DICTIONARY
     1         ',JSEQ)
  140 REWIND IUNIT
       WRITE(6,150)
  150  FORMAT('1')
      STOP
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
C     THIS VERSION OF PRIJOB IS LIMITED TO AN NCASE (MAXIMUM) OF...
C                        68,849
C
      INTEGER OPU
      DIMENSION NLAB(100)
      DIMENSION NMP1(26)
      DIMENSION SQID(13)
      DIMENSION LABEL(8,100),NMEMB(100),LAB(8)
      DIMENSION ANUMP (2026), AVE (2026), LTIT(100), NM58(1)
      DIMENSION RQID(100),HEADNG(14,3),ISEQ(100),ANMP(2026,2),
     1          INDEX(2026),INDEX1(2026),ANMP1(12)
      DIMENSION JA(50),AA(2),ITST(2),LTAB(2026),KTIT(465),OPU(2),KA(50)
C
      DATA AST / 3HEND/
      DATA IFIRST/0000001000001/
      DIMENSION KOMENT(7,2)
      DATA KOMENT/84HPERCENT OF MEMBERS PERFORMING              AVERAGE P
     1ERCENT TIME SPENT BY ALL MEMBERS /
      DATA PROGID/6HPRIJOB/,ITST/6HEND OF,6H DATA /,MBIT/262144/
      DATA IONE,IKAY/'1','0'/
      EQUIVALENCE (AA(1),JA(1))
      EQUIVALENCE (NM58(1), ANUMP(1))
C
      CALL SETUP(10,27,26,0,0,0,0,0,0,0,0,JSEQ)
      IHT = 10
      IJD = 27
      IRD=5
      IBT = 26
      IPR = 6
      OPU(1) = IPR
      OPU(2) = IBT
      IOT=0
      KPASS=1
      REWIND IHT
      CALL DISK(6000,465)
C
C
C     READ COMMUNICATION REGION AND CONTROL CARD
C
      READ(IHT) K,(JA(L),L=1,K)
      READ (IRD, 500) PROG, SID1, SID2, IPRINT
500   FORMAT (A6, X, A6, A2, X, I1)
      IF(SID1.EQ.AA(1).AND.SID2.EQ.AA(2)) GO TO 10
      WRITE(IPR,502) PROGID,AA(1),AA(2)
  502 FORMAT(31HOERROR IN CONTROL CARD. TAPE I5/1X,A6,1X,A6,A2/
     113HOJOB ABORTED.)
      CALL ERTRAN(2)
   10 CONTINUE
      NCASE=JA(3)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      NDUTY=JA(9)
      NTASK=JA(12)
C
C     READ OPTIONS CARD AND CHECK FOR INPUT ERRORS.
C
   60 CONTINUE
      KIDI=0
      NREQ=0
      READ(IRD,525,END=250) EXT,LAB,NHDG,IBCD,KSW,NPTSK,TMIN
  525 FORMAT(A6,1X,8A6,3I2,1X,I3,1X,F5.0)
      IBCD = 1 - IBCD
      NLN = 55
      IF(NHDG.LE.3) GO TO 62
      WRITE(IPR,503)
  503 FORMAT(55H0INCORRECT VALUE FOR 'NHDG'. MUST BE 0,1,2,3.JOB ABORT.)
      STOP
   62 CONTINUE
      IF(IBCD.GE.0) GO TO 64
      WRITE(IPR,504)
  504 FORMAT(55H0INCORRECT VALUE FOR 'IBCD'. MUST BE 0,1.    JOB ABORT.)
      CALL ERTRAN(2)
   64 CONTINUE
      IF(KSW.EQ.1.OR.KSW.EQ.2) GO TO 66
      WRITE(IPR,505)
  505 FORMAT(55H0INCORRECT VALUE FOR 'KSW'.  MUST BE 1,2.    JOB ABORT.)
      CALL ERTRAN(2)
   66 CONTINUE
      IF(NPTSK.LE.465) GO TO 67
      WRITE(IPR,506)
  506 FORMAT(55H0INCORRECT VALUE FOR 'NPTSK'.MUST BE 1-465.  JOB ABORT.)
      CALL ERTRAN(2)
   67 CONTINUE
      IF(NLN.GT.0) GO TO 68
      WRITE(IPR,526)
  526 FORMAT(55H0INCORRECT VALUE FOR 'NLN'.MUST BE NON-ZERO. JOB ABORT.)
   68 CONTINUE
      IF(NPTSK.EQ.0.AND.TMIN.EQ.0.) GO TO 72
      IF(NPTSK.GT.0.OR.TMIN.GT.0.)GO TO 71
   72 WRITE(IPR,527)
  527 FORMAT(85H0EITHER COLUMNS 63-65 OR COLUMNS 67-71 MUST BE NON-ZERO,
     1 BUT NOT BOTH.  JOB ABORT.         )
      CALL ERTRAN(2)
   71 CONTINUE
C
C     READ PAGE HEADINGS CARD(S).
C
      IF(NHDG.EQ.0) GO TO 77
      DO 76 L=1,NHDG
   76 READ(IRD,508) (HEADNG(I,L),I=1,14)
  508 FORMAT(13A6,A2)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
    77 CONTINUE
C      IF(KPASS.GT.1) GO TO 50
C
C      READ TASK/DUTY TITLES AND STORE ON DISK.
C      READ TITLE LOCATOR TABLE INTO LTAB ARRAY.
C
       NTRACK=0
       NTD=NTASK+NDUTY
    20 READ(IHT) K,LTAB(1),LTAB(2)
       IF(LTAB(1).NE.ITST(1).AND.LTAB(2).NE.ITST(2)) GO TO 20
    30 READ(IHT) K,(LTAB(L),L=1,K)
       IF(K.EQ.NTD.AND.LTAB(1).EQ.IFIRST) GO TO 40
       NTRACK=NTRACK+1
       CALL DSKSTO(465,LTAB(1),NTRACK,0)
       GO TO 30
    40 LTRAK1=NTRACK
C
C      READ REQUEST CARDS
C
    50 REWIND IJD
       NR=0
       NSKP=0
    51 NR=NR+1
       IF(NR.GT.100) GO TO 81
    55 NREQ=NREQ+1
    75 READ(IRD,507) SD,RQID(NR),(LABEL(L,NR),L=1,8),ISEQ(NR)
   507 FORMAT(A4,9A6,18X,I4)
       IF(SD.NE.AST) GO TO 51
80     CONTINUE
       NREQ=NREQ-1
    81 CONTINUE
       NR=NR-1
       IF(NR.EQ.0) GO TO 250
C
C      FIND MAXIMUM TAPE SEQUENCE NUMBER
C
       MAXSEQ=0
       DO 95 I=1,NR
       IF(ISEQ(I).LE.MAXSEQ) GO TO 95
       MAXSEQ=ISEQ(I)
    95 CONTINUE
C
C      SEARCH BINARY JOB DESCRIPTION TAPE FOR REQUESTED REPORTS.
C
   100 ISQ=0
       IREQ=0
       NFAIL = 0
102    READ(IJD) IDSP,NTD,(AVE(L),L=1,NTD),NR58,(NM58(L),L=1,NR58),
      1 ISQ, IR, (KA(L), L=1, IR), (AVE(L), L= 1, NTD),(ANUMP(L),L=1,NTD)
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      DO 150 II=1,NR
      IF(ISEQ(II).NE.ISQ) GO TO 150
      IREQ=IREQ+1
      MBRS=KA(17)
      NMEMB(II)=MBRS
      AMBRS=MBRS
      NZE=0
      DO 110 JJ=1,NTASK
      IF(ANUMP(JJ).NE.0..OR.AVE(JJ).NE.0.) NZE=NZE+1
      ANMP(JJ,1)=ANUMP(JJ)/AMBRS*100.
  110 ANMP(JJ,2)=AVE(JJ)/AMBRS
C
C     COMPUTE AND EXTRACT DESIRED NO. OF JOB IDENTIFIERS IN ORDER OF-
C     (1) PER CENT MEMBERS PERFORMING IF KSW=1
C     (2) AVERAGE PER CENT TIME SPENT BY ALL MEMBERS IF KSW=2
C
      IF(TMIN.GT.0.) GO TO 121
      NPTO=MIN0(NZE,NPTSK)
      DO 125 KK=1,NPTO
      XMAX=0.
      DO 120 LL=1,NTASK
      IF(ANMP(LL,KSW).LE.XMAX) GO TO 120
      XMAX=ANMP(LL,KSW)
      KEEP=LL
  120 CONTINUE
      ANMP(KEEP,KSW)=-ANMP(KEEP,KSW)
  125 INDEX1(KK)=KEEP
      GO TO 124
121   NPTO=0
      DO 123 KK=1,NTASK
       IF(ANMP(KK,KSW).EQ.0.) GO TO 123
       ANMP(KK,KSW)=ANMP(KK,KSW) + .01
      IF(ANMP(KK,KSW).LT.TMIN)GO TO 123
      NPTO=NPTO+1
      IF(NPTO.LE.465)GO TO 122
      WRITE(IPR,529)RQID(II),ISEQ(II)
  529 FORMAT(43H0NUMBER OF PRIMARY IDENTIFIERS FOR REPORT ',A6,17H', SEQ
     1UENCE NO. '14,35H' EXCEEDS 465. THIS REPORT SKIPPED.    )
       GO TO 102
122    INDEX1(NPTO)=KK
123    CONTINUE
       IF(NPTO.GT.0)GO TO 124
       NFAIL = NFAIL + 1
       IF (NFAIL .GT. 1) GO TO 10000
       WRITE (IPR, 20000)  IONE, IDSP, ISQ, TMIN
       GO TO 124
10000 WRITE (IPR, 20000) IKAY, IDSP, ISQ, TMIN
C
20000 FORMAT (A4, 'NO TASK VALUES FOUND IN REPORT (', A6, ') - SEQUENCE
     1', I4, ' WHICH SATISFY THE INPUT MINIMUM TASK VALUE (',F6.1,')',/,
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
     2 26X, 'PROCESSING WILL NONE-THE-LESS BE ATTEMPTED.')
C
C
C      BUILD ARRAY OF PRIMARY IDENTIFIERS, ELIMINATING DUPLICATES.
C
  124 CONTINUE
      IF(IREQ.EQ.1) GO TO 132
      IEND=IE
      DO 130 KK=1,NPTO
      DO 128 LL=1,IE
  128 IF(INDEX1(KK).EQ.INDEX(LL)) GO TO 130
      IEND=IEND+1
      INDEX(IEND)=INDEX1(KK)
  130 CONTINUE
      IE=IEND
      GO TO 136
  132 DO 133 KK=1,NPTO
  133 INDEX(KK)=INDEX1(KK)
      IE=NPTO
      IEND=NPTO
  136 CONTINUE
C
C      STORE COMPUTED TASK VALUES ON DISK IN REQUEST ORDER
C
      DO 137 KK=1,NPTO
      IDX=INDEX1(KK)
  137 ANMP(IDX,KSW)=ABS(ANMP(IDX,KSW))
      NTRACK=LTRAK1+5*(II-1)
      DO 138 KK=1,5
      NTRACK=NTRACK+1
      IB=465*(KK-1)+1
  138 CALL DSKSTO(465,ANMP(IB,KSW),NTRACK,0)
      ISEQ(II)=-ISEQ(II)
  150 CONTINUE
      NSKP=0
      IF(IREQ.EQ.NR.OR.ISQ.GE.MAXSEQ) GO TO 173
      GO TO 102
  173 CONTINUE
      IPTP=IBCD+1
      DO 174 KK=1,IPTP
      NPAGE=0
      NLINE=MBIT
      IOP=OPU(KK)
      DO 174 II=1,NR
      IF(NLINE.LE.NLN) GO TO 65
      NPAGE=NPAGE+1
      WRITE(IOP,514) (LAB(I),I=1,8),  EXT,NPAGE
      NLINE=2
      IF(NPAGE.GT.1) GO TO 65
      IF(NHDG.EQ.0) GO TO 176
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      WRITE(IOP,509) ((HEADNG(I,L),I=1,14),L=1,NHDG)
  509 FORMAT(1X,13A6,A2)
  176 CONTINUE
      WRITE(IOP,501)
  501 FORMAT(70H0****** REQUEST INDEX FOR PRIMARY JOB TYPE IDENTIFIERS P
     1ROGRAM ******  //)
      WRITE(IOP,530)
  530 FORMAT(3H0NO,2X,6H IDSP ,12X,11HDESCRIPTION /)
      NLINE=NLINE+NHDG+4
   65 CONTINUE
      NLINE=NLINE+2
  174 WRITE(IOP,510) II,RQID(II),(LABEL(JJ,II),JJ=1,8),NMEMB(II)
  510 FORMAT(1H0,I2,2X,A6,2X,8A6,12H  CONTAINING,I5,9H MEMBERS.  )
      IF(IREQ.EQ.NR) GO TO 170
C
C     FLAG REQUESTED REPORTS WHICH WERE NOT FOUND ON J.D. TAPE
C
  160 DO 165 II=1,NR
      IF(ISEQ(II).LT.0) GO TO 165
      NSKP=NSKP+1
      WRITE(IPR,516) RQID(II),ISEQ(II)
  516 FORMAT(21H0CANNOT FIND REPORT ',A6,19H',TAPE SEQUENCE NO.,I4,
     123H.  THIS REPORT SKIPPED. )
      ANMP(1,1)=-1.
      NTRACK=LTRAK1+5*II-4
      CALL DSKSTO(465,ANMP(1,1),NTRACK,0)
  165 CONTINUE
  170 NCOMP=NR-NSKP
      IF (NFAIL .LT. NCOMP) GO TO 10200
10150 WRITE (IPR, 20010)  NFAIL
20010 FORMAT ('K', 100('*'), /, '  ALL OF THE ', I4, ' REPORTS REQUESTED
     1 FAILED TO HAVE ANY TASKS WHICH SATISFIED THE INPUT MINIMUM TASK V
     2ALUE.')
C
      GO TO 60
C
10200 WRITE (IPR, 513) NCOMP, NSKP
C
  513 FORMAT(55H0PRIMARY IDENTIFIER TASK NUMBERS HAVE BEEN COMPUTED FOR,
     1I3, 9H REPORTS.,I3,37H REPORTS NOT FOUND ON JOB DESC. TAPE. )
C
C     SORT PRIMARY IDENTIFIERS IN TASK ORDER
C
      NPTO=IEND
      DO 145 KK=1,NPTO
      MMIN=NTD
      DO 140 LL=1,NPTO
      IF(INDEX(LL).GE.MMIN) GO TO 140
      MMIN=INDEX(LL)
      KEEP=LL
```

LINE # TEXT...

```
  140 CONTINUE
      INDEX1(KK)=INDEX(KEEP)
  145 INDEX (KEEP)=NTD
C
C     FORM REPORTS IN REQUEST ORDER, OUTPUT TO PRINTER
C     AND BCD REPORT TAPE IF IBCD IS NON-ZERO.
C
      IPASS=1
      LTRAK=LTRAK1
      KNT=0
      NLINE=MBIT
      MINI=6
      IF(KSW.EQ.1) MINI=13
175   LMT=MIN0(MINI,NR-KNT)
      ILM=0
      DO 200 II=1,NPTO
      IDX=INDEX1(II)
      LM=LTAB(IDX)/MBIT
      LN=MOD(LTAB(IDX),MBIT)
      IF(ILM.NE.LM) CALL DSKRED(465,KTIT(1),LM,0)
      ILM=LM
      LM=KTIT(LN)/MBIT
      LE=MOD(KTIT(LN),MBIT)
      NW=LM-LE
      DO 190 KK=1,NW
      LKK=LN+KK
  190 LTIT(KK)=KTIT(LKK)
C
      NTRACK=LTRAK
      KID=KID1
      DO 192 JJ=1,LMT
      NTRACK=NTRACK+1
      IB=1
  182 CONTINUE
      CALL DSKRED(465,ANMP(1,1),NTRACK,0)
      KID=KID+1
      IF(ANMP(1,1).GE.0.)GO TO 181
      NTRACK=NTRACK+5
      GO TO 182
  181 CONTINUE
      SQID(JJ)=RQID(KID)
      NLAB( JJ)=KID
      DO 185 KK=1,4
      IB=IB+465
      NTRACK=NTRACK+1
      NREAD=465
      IF(KK.EQ.4) NREAD=166
  185 CALL DSKRED(NREAD,ANMP(IB,1),NTRACK,0)
  192 ANMP1(JJ)=ANMP(IDX,1)
      IF ((IPASS .GT. 1) .AND. (IPRINT .EQ. 1)) GO TO 195
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
C     FIRST PASS, WRITE COMPLETE TASK TITLES, WITH 6 JOB DESCRIPTIONS.
C
      IF(NLINE.LE.NLN) GO TO 191
      NPAGE=NPAGE+1
      DO 260 JJ=1,IPTP
      IOP=OPU(JJ)
       WRITE(IOP,514) (LAB(I),I=1,8),  EXT,NPAGE
  514 FORMAT(1H1,8A6,29X,,A6,6H PAGE ,I3)
      IF(NPAGE.GT.2) GO TO 189
      WRITE(IOP,517) (KOMENT(L,KSW),L=1,7)
  517 FORMAT(42H0PRIMARY JOB TYPE IDENTIFIERS SELECTED BY ,7A6/)
      IF(NHDG.EQ.0) GO TO 189
      DO 188 L=1,NHDG
  188 WRITE(IOP,509) (HEADNG(I,L),I=1,14)
  189 CONTINUE
      IF(KSW.EQ.1)GO TO 274
      WRITE(IOP,515) (SQID(L),L=1,LMT)
       WRITE(IOP,934)
  934  FORMAT(1H0)
  515 FORMAT(1H0,6HDY-TSK,24X,11HDESCRIPTION,25X,6(4X,A6))
      GO TO 260
 274   WRITE(IOP,531)(NLAB(L),L=1,LMT)
       WRITE(IOP,934)
 531   FORMAT(1H0,6HDY-TSK,24X,11HDESCRIPTION,25X,13(2X,I3))
  260 CONTINUE
      NLINE=6+NHDG
  191 DO 265 JJ=1,IPTP
      IOP=OPU(JJ)
      IF(KSW.EQ.1)GO TO 275
      WRITE(IOP,518) (LTIT(L),L=1,11),(ANMPI(L),L=1,LMT)
  518 FORMAT(1H ,11A6,6F10.3)
      GO TO 265
 275  DO 276 L=1,LMT
  276 NMPI(L)=ANMPI(L)+.5
      WRITE(IOP,532)(LTIT(L),L=1,11),(NMPI(L),L=1,LMT)
  532 FORMAT(1H ,11A6,13(2X,I3))
 265  CONTINUE
      NLINE=NLINE+NW/11
      IF(NW.LE.11) GO TO 200
      DO 268 JJ=1,IPTP
      IOP=OPU(JJ)
  268 WRITE(IOP,522) (LTIT(L),L=12,NW)
  522 FORMAT(1X,11A6)
      GO TO 200
C
C     AFTER INITIAL PASS, WRITE TASK IDENTIFIERS, WITH 12 JOB DESC.
C
  195 CONTINUE
      IF(NLINE.LE.NLN) GO TO 196
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          NPAGE=NPAGE+1
          DO 270 JJ=1,IPTP
          IOP=OPU(JJ)
           WRITE(IOP,514) EXT,NPAGE
          IF KSW.EQ.1)GO TO 278
     *    WRITE(IOP,523) (SQID(L),L=1,LMT)
      523 FORMAT(1H0,6HDY-TSK,12(4X,A6)//)
           WRITE(IOP,934)
           GO TO 270
  278     WRITE(IOP,533)(NLAB(L),L=1,LMT)
  533     FORMAT(1H0,6HDY-TSK,25(2X,I3)//)
           WRITE(IOP,934)
      270 CONTINUE
          NLINE=3
      196 CONTINUE
          DO 272 JJ=1,IPTP
          IOP=OPU(JJ)
          IF(KSW.EQ.1)GO TO 280
          WRITE(IOP,524) LTIT(1),(ANMPI(L),L=1,LMT)
      524 FORMAT(1H ,A6,12F10.3)
          GO TO 272
  280     DO 281 L=1,LMT
  281     NMPI(L)=ANMPI(L)+.5
          WRITE(IOP,534) LTIT(1),(NMPI(L),L=1,LMT)
  534     FORMAT(1H ,A6,25(2X,I3))
      272 CONTINUE
          NLINE=NLINE+1
      200 CONTINUE
          KNT=KNT+LMT
          IF(KNT.GE.NR) GO TO 210
          NLINE=MBIT
          IF (IPRINT .NE. 1) GO TO 205
          IPASS=IPASS+1
          MINI=12
          IF(KSW.EQ.1)MINI=25
  205     LTRAK=LTRAK+5*LMT
          KIDI=KIDI+LMT
          GO TO 175
      210 CONTINUE
C
C         TRANSFER TO HERE AT END OF EACH REPORT WHEN 'LAST' CARD IS READ.
C
          IF(IBCD.NE.0) CALL REPEND(SIDI,EXT,LAB,JSEQ)
          REWIND IJD
          REWIND IMT
          KPASS=KPASS+1
          GO TO 60
C
C         TRANSFER TO HERE TO END PROGRAM WHEN 'END' CARD IS READ.
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
   250 CONTINUE
  290   CONTINUE
  C
        WRITE(6,377)
  377   FORMAT('1')
        STOP
  C
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          DIMENSION LCOMUN(1000),LRELOC(1000),ISAVE(30)
          DIMENSION IOP(5), LOC(22)
          COMMON /INFILE/ IDPROG, NAME(3)
           DIMENSION KSAVE(5)
           DIMENSION IOPRS(15)
           EQUIVALENCE(IOPEN,IOPRS(1))
           EQUIVALENCE(IREAD,IOPRS(2))
           EQUIVALENCE(IDSPR,IOPRS(3))
           EQUIVALENCE(ICOLL,IOPRS(4))
           EQUIVALENCE(IWRIT,IOPRS(5))
           EQUIVALENCE(ICLSE,IOPRS(6))
           EQUIVALENCE(IENDO,IOPRS(7))
           EQUIVALENCE(IAST,IOPRS(8))
           EQUIVALENCE(IDEL,IOPRS(9))
           DATA JREAD/6HREAD  /
           DATA JWRIT/6HWRITE /
           DATA IOPEN/6H/OPEN /
           DATA IREAD/6H/READ /
           DATA IDSPR/6H/DISPE/
           DATA ICOLL/6H/COLLE/
           DATA IWRIT/6H/WRITE/
           DATA ICLSE/6H/CLOSE/
           DATA IENDO/6H/END  /
           DATA IAST/1H /
           DATA IDEL/6H/AUDIT/
           DATA IOPRS(10)/4H/XQT/
           DATA K99997/4HC008/
           DATA IBLNKS/6H      /
           DATA MV/1HV/
           DATA MC/1HC/
           DATA MN/1HN/
           DATA MT/1HT/
           DATA ISLSH/1H//
           DATA IDL/1HL/
           DATA ID99/6HKC9999/
           DATA IDLI/1HI/
           DATA LD/5HLDICT/
           DATA IC/5HICEES/
           DATA IN/5HIEES/
           DATA IDLN/1HN/
           DATA KSAVE/30HPROGRAM GENERATION PROGRAM       /
           DATA IUSER /'USER'/
          DATA ICSAVE/'000'/
  900     FORMAT(7X,A6,A2,4(1X,A2))
  901      FORMAT(A1,A3,2X,12A6)
  902      FORMAT(8X,15HEQUIVALENCE(KC(,I4,3H),C,J3,     1H),50X)
  903      FORMAT(8X, 9HINTEGER C,J3,64X)
  904      FORMAT(8X,'DIMENSION TASKS(2000), T(2000)')
  905      FORMAT(8X,13HINTEGER TASKS ,63X)
  906      FORMAT(8X,35HDIMENSION KC(500),KV(2000),KT(2000)   ,39X)
```

```
LINE #   TEXT...
  908    FORMAT(8X,34HDIMENSION LCOMUN(50),LRELOC(1000)   ,42X)
  911   FORMAT(8X,'REWIND 10',/,8X12HDATA IC/'C'/)
  912    FORMAT(8X,19HKCVARS=KCVARS+NCOMP,57X)
  913   FORMAT(7X,6HDATA I,A1,A1,A3,6H  /90H,A1,A3,2H =,7A6,8X/5X,1H1,4A6,
       1      12X,A6,17X,1H/,14X)
  914    FORMAT(7X,11HDIMENSION I,A1,A1,A3,4H(15),57X)
  915    FORMAT(7X,13HEQUIVALENCE(I,A1,6HEES(1,,I3,3H),I,A1,A1,A3,4H(1)),
       1  42X)
  916    FORMAT(7X,11HDIMENSION I,A1,7HEES(15,,I3,1H),54X)
  917    FORMAT(7X, 9HINTEGER V,J3,65X)
  918    FORMAT(7X,15HEQUIVALENCE(KV(,I4, 3H),V,J3,1H),51X)
  919    FORMAT(7X,15HEQUIVALENCE(KV(,I4, 3H),V,J3,4H(1)),48X)
  920    FORMAT(7X,11HDIMENSION V,J3,1H(,I3,1H),58X)
  921    FORMAT(7X,13HDATA   ID1/6H,A6,10H/,  ID2/2H,A2,1H/,45X)
  922    FORMAT(7X, 6HNCOMP=,I3,68X)
  923    FORMAT(7X,30HREAD(10) K,(LCOMUN(I),I=1,K)      ,47X)
  924    FORMAT(7X,20HKCASE=LCOMUN(NDICT)  ,57X)
  925    FORMAT(7X,20HNSELCT=LCOMUN(NCASE) ,57X)
  926    FORMAT(8X,26HLCOMUN(NDICT)=KCASE+NCOMP+,I4,47X)
  927    FORMAT(7X,46HIF((ID1.EQ.LCOMUN(1)).AND.(ID2.EQ.LCOMUN(2)))   ,
       1        12HGO TO 99998 ,19X)
  928    FORMAT('C STUDY-ID MISMATCH - RUN ABORTS.')
  929    FORMAT('99999  WRITE(6,99997)  LCOMUN(1), LCOMUN(2)')
  930    FORMAT('99997  FORMAT(27H STUDY-ID MISMATCH; TAPE = ,A6,A2)')
  931    FORMAT(7X,'CALL ERTRAN(2)')
  932    FORMAT(36H99998  WRITE(15) K,(LCOMUN(I),I=1,K) ,   48X)
  933    FORMAT(8X,44HREAD (10)K,(LRELOC(I),I=1,K),L,(KV(I),I=1,L),33X)
  934    FORMAT(8X,44HWRITE(15)K,(LRELOC(I),I=1,K),L,(KV(I),I=1,L),33X)
  935    FORMAT(8X, 6HISW=0 ,70X)
  936    FORMAT(8X,18HDO 99996 I=1,KCASE ,58X)
  937    FORMAT(8X,47HREAD(10)KC9999,((LDICT(K,J),K=1,15),J=1,KC9999),29X)
  938   FORMAT(8X,'DECODE(99982,LDICT(1,1)) ICHAR ',/,8X,'IF(ICHAR.EQ.IC)
       1GO TO 99995',/,'99982   FORMAT(1A1)')
  939    FORMAT(8X,24HIF(ISW.EQ.1) GO TO 99995,52X)
  940    FORMAT(8X, 3HM=1,73X)
  941    FORMAT(8X,18HDO 99994 L=1,NCOMP,58X)
  942    FORMAT(8X,30HWRITE(15)M,(ICEES(N,L),N=1,15)  ,46X)
  943    FORMAT(15H99994 CONTINUE ,69X)
  944    FORMAT(8X, 6HISW=1 ,70X)
  945    FORMAT(45H99995   WRITE(15)KC9999,((LDICT(K,J),K=1,15),  ,
       1          11HJ=1,KC9999),29X)
  946    FORMAT(15H99996  CONTINUE ,69X)
  947    FORMAT( 8X,21HDIMENSION LDICT(15,1),55X)
  948    FORMAT( 8X,30HEQUIVALENCE(LDICT(1,1),KC(1))  ,46X)
  949    FORMAT( 8X, 7HNDICT=4,69X)
  950    FORMAT( 8X, 7HNCASE=3,69X)
  951    FORMAT( 8X, 16HNTASK=LCOMUN(12),60X)
  952    FORMAT( 8X,23HDO 99993 KPATH=1,NSELCT,53X)
  953    FORMAT(8X,A5,1H(,I2,34H)KCVARS,(KC(I),I=1,KCVARS),KVVARS,  ,
       1 19H(KV(I),I=1,KVVARS),,16X/5X,1H1,2X,28HKTASKS,(TASKS(I),I=1,KTAS
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
       1KS),48X)
954     FORMAT(8X,19HDO 99992 I=1,NTASK   ,/,8X,'T(I) = 0.0 ')
955     FORMAT(15H99992   KT(I)=0,69X)
956     FORMAT(8X,19HDO 99991 I=1,KTASKS ,57X)
957     FORMAT(8X,17HJ=TASKS(I)/ISHF18,59X)
958     FORMAT(8X,22HS=MOD(TASKS(I),ISHF18),/,8X,'T(J) = S/1000.')
959     FORMAT(8X,25HKT(J)=TSKTOT*S/100000.+.5,51X)
960     FORMAT(16H99991    CONTINUE,68X)
961     FORMAT(8X, 8HKTASKS=0,68X)
962     FORMAT(8X,19HDO 99990 I=1,NTASK ,57X)
963     FORMAT(8X,25HIF(KT(I).EQ.0)GO TO 99990,51X)
964     FORMAT(8X,15HKTASKS=KTASKS+1,61X)
965     FORMAT(8X,31HTASKS(KTASKS)=S/TSKTOT          )
966     FORMAT(15H99990  CONTINUE,69X)
967     FORMAT(8X,13HEQUIVALENCE(T,J4,4H,KT(,I4,'))')
968     FORMAT(8X, 9HINTEGER T,J4,63X)
969     FORMAT(8X,20HDATA ENDOF/6HEND OF/,56X)
970     FORMAT(8X,13HINTEGER ENDOF,63X)
971     FORMAT(8X, 5HISW=0 ,71X)
972     FORMAT(31H99989  READ(10) K,(KV(I),I=1,K) ,54X)
973     FORMAT(8X,24HWRITE(15)K,(KV(I),I=1,K),53X)
974     FORMAT(8X,46HIF((KV(1).EQ.ENDOF).AND.(ISW.EQ.1))GO TO 99988,30X)
975     FORMAT(8X,24HIF(KV(1).EQ.ENDOF) ISW=1 ,52X)
976     FORMAT(8X,11HGO TO 99989 ,65X)
977     FORMAT(    15H99993  KC9999=0,69X)
978    FORMAT(13A6,A2)
979     FORMAT(8X,12HISHF18=2**18,64X)
980     FORMAT(10H99988  I=0,74X)
981     FORMAT(7X,3HEND,74X)
984     FORMAT(8X,17HDIMENSION ICMPRE(,I3,1H),55X)
985     FORMAT(8X,6HDATA I,A1,1HM,A3,3H/6H,A1,A3,3H =/,55X)
986     FORMAT(8X,19HEQUIVALENCE(ICMPRE(,I3,3H),I,A1,1HM,A3,1H),45X)
987     FORMAT(8X,19HDO 99987 K=1,KC9999,57X)
988     FORMAT(8X,12HJ=LDICT(1,K),64X)
989     FORMAT(8X,13HDO 99986 N=1,,I3,60X)
990     FORMAT(8X,29HIF(J.NE.ICMPRE(N))GO TO 99986 ,47X)
991     FORMAT(8X,15HDO 99985 M=1,15 ,61X)
992     FORMAT(30H99985   LDICT(M,K)=IVEES(M,N)    , 54X)
993     FORMAT(8X,11HGO TO 99987,65X)
994     FORMAT(16H99986   CONTINUE,68X)
995     FORMAT(16H99987   CONTINUE,68X)
996     FORMAT(12H99973   STOP,72X)
997     FORMAT(8X, 7HTSKTOT=,A4,65X)
998     FORMAT(8X,  3HJ=0,73X)
999     FORMAT(8X, 9HJ=J+KT(I),57X)
1000    FORMAT(8X,14HS=KT(I)*100000,52X)
1001    FORMAT(8X,A4,2H=J,70X)
1002    FORMAT(8X,11HC007=KTASKS,65X)
1003    FORMAT(8X,11HC002=KTASKS,65X)
1004    FORMAT(A1,A5,13A6)
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
1005   FORMAT(A1,A3, 2X,I2)
1006    FORMAT(8X,17HDIMENSION ICXXXX(,I4,1H) ,54X)
1007    FORMAT(8X, 8HDATA ICX,J3,1H/,I8,1H/,55X)
1008    FORMAT(8X,19HEQUIVALENCE(ICXXXX(,I4,5H),ICX,J3,1H),44X)
1009    FORMAT(8X,'M = 1',71X,/,8X,13HDO 99984 L=1,,I4,59X)
1010    FORMAT(8X,31HWRITE(15) M,(INEES(I,L),I=1,15) ,46X)
1011    FORMAT(16H99984  CONTINUE  ,68X)
1012    FORMAT(8X, 4HK=K+,I4,68X)
1013    FORMAT(8X, 7HKVVARS=,I4,65X)
1014    FORMAT(I2,14(1X,A1,I3))
1015    FORMAT(8X,11HICMPRE(N)=0,65X)
1019    FORMAT(8X,15HEQUIVALENCE(KC(,A3,3H),C,A3,      1H),50X)
1020    FORMAT(8X,43HREAD (10)K,(LRELOC(I),I=1,K)                   ,33X)
1021    FORMAT(8X,3HL=1,73X)
1022   FORMAT('99983  FORMAT(1X,15A6)')
1023    FORMAT(8X,15HWRITE(6,99983)(,A6,13H(K,J),K=1,15),52X)
1024   FORMAT('        CALL SETUP(10,15,11,12,16,17,20,0,0,0,0,JSEQ)')
1025   FORMAT('        CALL MARGIN')
1027   FORMAT(7X,'ICHEK = 1')
1029    FORMAT(8X,18HDO 99980 I=1,NTASK,58X)
1030    FORMAT(17H99980  N=N+KT(I) ,67X)
1031    FORMAT(8X,3HN=0,73X)
1032    FORMAT(8X,24HIF(N.EQ.0) GO TO 99979    ,52X)
1033    FORMAT(10H99979  I=1,74X)
1034    FORMAT(8X, 8HINRJCT=0,68X)
1035    FORMAT(8X, 8HINDFLG=0,68X)
1036    FORMAT(8X,17HDIMENSION INDXXX(,I4,1H),54X)
1037    FORMAT(14A5)
1038    FORMAT(8X, 8HDATA IND,J3, 3H/6H,A6,1H/,55X)
1039    FORMAT(8X,19HEQUIVALENCE(INDXXX(,I4,5H),IND,J3,1H),44X)
1040    FORMAT(8X,13HDO 99976 I=1,,I4,59X)
1041    FORMAT(8X,34HIF(KV(I).NE.INDXXX(I))GO TO 99976  ,42X)
1042    FORMAT(8X,13HIF(INRJCT.NE.,I3,22H)WRITE(6,99000) INRJCT,38X)
1044    FORMAT(8X,21HWRITE(6,99978) KV(I) ,55X)
1046    FORMAT(8X, 5HINZ=0,71X)
1047    FORMAT(8X,15HIF(N.EQ.0)INZ=1,61X)
1048    FORMAT(8X,32HIF(INZ.EQ.1)WRITE(6,99977) KV(I),44X)
1049    FORMAT(64H99977 FORMAT(1X,5HCASE ,A4,31H HAS NO NON-ZERO TASK RES
       1PONSES  ),18X)
1050    FORMAT(8X,21HINRJCT=INRJCT+1       ,55X)
1051    FORMAT(8X,21HGO TO 99993           ,55X)
1052    FORMAT(15H99976  CONTINUE,69X)
1053    FORMAT(8X,29HLCOMUN(NCASE)=LCOMUN(NCASE)-(,I4,1H),42X)
1054    FORMAT(8X,13HDO 99975 I=1,,I4,59X)
1055    FORMAT(8X, 5HJ=K+I,71X)
1056    FORMAT(26H99975  LRELOC(J)=ICXXXX(I)  ,58X)
1057    FORMAT(1HC,83X/1HC,83X/1HC,2A6,71X/1HC,83X/1HC,83X)
1058    FORMAT(48H W A R N I N G - LENGTH OF NEW VARIABLE WAS ZERO,36X)
1059    FORMAT(8X,36HTASKS(KTASKS)=TASKS(KTASKS)+I*ISHF18,40X)
1060    FORMAT(8X,26HVARIABLE LENGTH SET TO '1')
```

```
LINE #   TEXT...

 1061   FORMAT(8X,8HTSKTOT=N,68X)
 1062   FORMAT(8X,21HDATA YES/1H1/,NO/1H0/,55X)
 1063   FORMAT(8X,11HINTEGER YES,65X)
 1064   FORMAT(8X, 7HINMBR=0,69X)
 1065   FORMAT(8X,13HINMBR=INMBR+1,63X)
 1066   FORMAT(8X,10HC001=INMBR,66X)
 1067     FORMAT(1X,15A6)
 1068    FORMAT(8X,3HDO ,I4,3H J=,A1,1H,,A6,58X)
 1069   FORMAT(I4,4X,8HCONTINUE,59X)
 1071   FORMAT(8X,21HIF(ISW.EQ.0)GO TO 991,55X)
 1072   FORMAT(16H992      CONTINUE,68X)
 1073   FORMAT(8X,9HGO TO 992,67X)
 1074   FORMAT('991',5X,'IF(K.EQ.2.AND.KV(1).EQ.ENDOF) GO TO 993',
       A /,8X,'WRITE(6,994) NSELCT',/,'994',5X,'FORMAT(46H RUN HALTS - END
       B OF DATA NOT FOUND AFTER CASE  ,I5)',/,8X,'CALL ERTRAN(2)',/,
       C '993',5X,'CONTINUE',/,8X,'KV(3) = 1',/,
       1 8X, 'KV(4) = 0',  67X, /,
       2 8X, 'KV(5) = 1',  67X, /,
       3 8X, 'KV(6) = 0',  67X, /,
       4 8X, 'WRITE (15) K, (KV(I), I=1,6)',  49X)
 1075   FORMAT(8X,9HLCOUNT=99,67X)
 1076  FORMAT(6X,'IF(MOD(KPATH,ICHEK).NE.0) GO TO 99804',/,6X,'IF(KPATH.G
       1E.5) ICHEK = 250',/,'99800 WRITE(6,99801) KPATH, C001, KV(1)',/,
       2'99801 FORMAT(37H0 TAPE SEQ    CASE-ID     CASE CONTROL,/,6X,I4,',/
       35X,'17X,I4,12X,A4,/ ,29H CXXX VARIABLES - 10 PER LINE)',/,6X,
       4'WRITE(6,99802) (KC(I),I=1,KCVARS)',/,'99802 FORMAT(3(7X,I6),
       57F13.3,/,10(F13.3))',/,6X,'WRITE(6,99803) (KV(I),I=1,KVVARS)',/,
       6'99803 FORMAT(35H VXXX VARIABLES - 20 WORDS PER LINE,/'/5X,'1122H
       7/  1  /  2  /  3  /  4  /  5  /  6  /  7  /  8  /  9  /  10 /',/,
       85X,'2  11 /  12 /  13 /  14 /  15 /  16 /  17 /  18 /  19 /  20 /,
       9/,',/,5X,'3(1X,20A6))',/,'99804 CONTINUE')
 1077   FORMAT(12A6,A4,A4)
 1078   FORMAT(7X,'NVXXX = ',I3)
 1079  FORMAT(7X,'ENDFILE 15',/,7X,'IUNIT = ',I2)
 1080  FORMAT(7X,'REWIND IUNIT',/,7X,'READ(IUNIT) I, I, I, I, KCASE',/,
       17X,'READ(IUNIT)',/,7X,'CALL PRTDIC(IUNIT,6,0,IDUM,KCASE)'/,
       27X,'CALL MARGIN',/,7X,'KT(1) = 0',/,7X,'T(1) = 0.0')
C
C              THIS CONSTANT IS USED FOR SHIFTING, NOT PACKING
C
        ISHF18=2**18
       CALL SPREP(IOP)
        IFLAG = 0
        IPRT = 0
        IF(FLD(6,6,IOP(1)).EQ.1HN) IPRT = 1
       CALL REWIND
C
       CALL SETUP(10,0,0,0,0,0,28,0,0,0,0,JSEQ)
       CALL MARGIN
       IF(IPRT.EQ.0) CALL RESET
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      IF(IPRT.EQ.0) CALL ERRMSG(4,'(PHASE 1 CONTROL CARDS)')
C
C
C               READ CONTROL CARD
C
C
      READ(5,900) ID1, ID2, NCOMP, NEWVAR, NC, NVARS
      IF(IPRT.EQ.0) CALL PRINTC
      CALL BCDBIN(NCOMP , NCOMP,2,$1026)
      CALL BCDBIN(NVARS , NVARS,2,$1026)
      CALL BCDBIN(NEWVAR,NEWVAR,2,$1026)
      CALL BCDBIN(    NC,     NC,2,$1026)
      JOPN = 1
C
C
C               READ COMMUNICATIONS REGION
C
C
  100  READ(10) N,(LCOMUN(I),I=1,N)
       KCASE=LCOMUN(4)
       JSID1=LCOMUN(1)
       JSID2=LCOMUN(2)
       NTASK=LCOMUN(12)
       IF((ID1.EQ.JSID1).AND.(ID2.EQ.JSID2)) GO TO 120
      IF(ID1.EQ.' '.AND.ID2.EQ.' ') GO TO 1200
      WRITE(6,661) ID1,ID2, JSID1,JSID2
  661 FORMAT(' STUDY-ID MISMATCH: CARD = ',A6,A2,' TAPE = ',A6,A2)
       GO TO 99999
 1200 ID1 = JSID1
      ID2 = JSID2
  120  CONTINUE
C
C
C               ZAP OUT IBFTC,COMMON AND DIMENSION STATEMENTS
C
C
       WRITE(28,906)
       WRITE(28,905)
       WRITE(28,904)
       WRITE(28,908)
C
C
C               ZAP OUT FIRST THREE COMPUTED VARIABLES
C
C
       DO 102  I=1,3
       WRITE(28,902)I,I
       WRITE(28,903) I
  102  CONTINUE
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
C              ZAP OUT COMPUTED VARIABLES C004-C007
C
C
      DO 103 I=4,7
      WRITE(28,902) I,I
  103 CONTINUE
C
C
C              BUILD DATA STATEMENTS,EQUIVALENCES,ETC FOR NEW COMPUTED
C              VARIABLES AND DICTIONARY CHANGES. ONE SET OF CODING FOR
C              BOTH. ONLY THE CONTROL WORDS KTYPE,N AND K CHANGE.
C
C
      KTYPE=MC
      N=NCOMP
      K=0
  113 IF(N.EQ.0) GO TO 112
      WRITE(28,916) KTYPE,N
      IF(K.EQ.1) WRITE(28,984) N
      DO 104 J=1,N
      IF(K.EQ.0) GO TO 114
      READ(5,901)JTYPE,JNUM
     IF(IPRT.EQ.0) CALL PRINTC
      WRITE(28,985) JTYPE,JNUM,JTYPE,JNUM
      WRITE(28,986) J,JTYPE,JNUM
  114 READ(5,901) ITYPE,NUM,(ISAVE(I),I=1,12)
     IF(IPRT.EQ.0) CALL PRINTC
      IF(ISAVE(12).EQ.IBLNKS)WRITE(6,140) ITYPE,NUM
      IF(ITYPE.NE.MC.AND.ITYPE.NE.MV) GO TO 138
      IF(ITYPE.EQ.MV.AND.K.EQ.0) GO TO 137
      IF(JOPN.EQ.2.AND.ITYPE.EQ.MC) GO TO 136
      IF(K.EQ.0.AND.ITYPE.EQ.MC)ICSAVE=NUM
      IF(K.EQ.0)WRITE(28,1019)NUM,NUM
      WRITE(28,914)KTYPE,ITYPE,NUM
      WRITE(28,913) KTYPE,ITYPE,NUM,ITYPE,NUM,(ISAVE(I),I=1,12)
      WRITE(28,915) KTYPE,J,KTYPE,ITYPE,NUM
  104 CONTINUE
  112 I=0
      K=K+1
      N=NVARS
      KTYPE=MV
      IF((K.EQ.1).AND.(NVARS.GT.0)) GO TO 113
      KTYPE=MN
C
C
C              READ RELATIVE LOCATION TABLE
C
C
      READ(IU) KA,(LRELOC(I),I=1,KA)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          KRELOC=LRELOC(KA)/ISHF18+MOD(LRELOC(KA),ISHF18)
C
C                  BUILD DATA STATEMENT, EQUIVALENCES FOR NEW BACKGROUND VAR
C
C
C
          IF(NEWVAR.EQ.0) GO TO 124
          WRITE(28,916) KTYPE,NEWVAR
          WRITE(28,1006) NEWVAR
          DO 121  J=1,NEWVAR
          READ(5,1005)ITYPE,NUM,LEN
          IF(IPRT.EQ.0) CALL PRINTC
           IF(LEN.EQ.0)LEN=1
          READ(5,901) JTYPE,INUM,(ISAVE(I),I=1,12)
          IF(IPRT.EQ.0) CALL PRINTC
           IF(ITYPE.NE.JTYPE)GO TO 99998
           IF(NUM.NE.INUM) GO TO 99998
           IF(ITYPE.NE.MV)GO TO 139
           IF(ISAVE(12).EQ.IBLNKS)WRITE(6,140) ITYPE,NUM
          CALL BCDBIN(INUM,LNUM,3,$1100)
          IF(LNUM.NE.(KA+J)) GO TO 139
           WRITE(28,914)KTYPE,ITYPE,NUM
           WRITE(28,913) KTYPE,ITYPE,NUM,ITYPE,NUM,(ISAVE(I),I=1,12)
           WRI.E(28,915) KTYPE,J,KTYPE,ITYPE,NUM
           M= LEN*ISHF18+KRELOC
           WRITE(28,1007) J,M
           WRITE(28,1008) J,J
           WRITE(28,917) LNUM
           L=KRELOC
           IF(LEN.GT.1) GO TO 122
           WRITE(28,918) L,LNUM
           GO TO 123
  122      WRITE(28,920) LNUM,LEN
           WRITE(28,919) L,LNUM
  123      KRELOC=KRELOC+LEN
  121      CONTINUE
  124      KRELOC=KRELOC-1
C
C
C                  BUILD CONTROL DICTIONARY FOR REFERENCING VARIABLE NAMES
C
C
C
C
           DO 105 J=1,KA
           IF(NC.GT.0)GO TO 105
           NUM=LRELOC(J)/ISHF18
           IDISP=MOD(LRELOC(J),ISHF18)
           WRITE(28,917) J
           IF(NUM.GT.1)GO TO 106
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
            WRITE(28,918) IDISP,J
            GO TO      105
  106       WRITE(28,920)          J,NUM
            WRITE(28,919) IDISP,J
  105       CONTINUE
C
C
C                 BUILD CONTROL DICTIONARY FROM STATEMENT CARD
C
C
           LCOUNT=99
           IF(NC.EQ.0) GO TO 129
           DO 126 I=1,NC
           READ(5,1014) K,(ISAVE(J),LCOMUN(J),J=1,K)
          IF(IPRT.EQ.0) CALL PRINTC
           K=IABS(K)
           DO 126 J=1,K
           IF(K.EQ.0)GO TO 126
           IF(ISAVE(J).EQ.MV) GO TO 127
           IF(ISAVE(J).NE.MT.AND.ISAVE(J).NE.'1') GO TO 141
          N = LCOMUN(J)
          IF(ISAVE(J).EQ.'1') N = N + 1000
           WRITE(28,967) N,N
           WRITE(28,968) N
           GO TO 126
  127      N=LCOMUN(J)
          L = N
           WRITE(28,917) L
           M=MOD(LRELOC(N),ISHF18)
           N=LRELOC(N)/ISHF18
           IF(N.GT.1) GO TO 128
           WRITE(28,918) M,L
           GO TO 126
  128      WRITE(28,920) L,N
           WRITE(28,919) M,L
  126      CONTINUE
  129      CONTINUE
C
C
C            READ DICTIONARY AND PRINT
C
C
          LCOUNT = 99
          CALL PRTDIC(10,6,0,IDUM,KCASE)
          IF(IPRT.EQ.0) CALL RESET
          IF(IPRT.EQ.0) CALL ERRMSG(4,'(PHASE 2 CONTROL CARDS)')
C
C
C                READ FIRST DATA RECORD AND CHECK NUMBERING OF NEW CXXX'S
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C
      READ(10)K,(LCOMUN(I),I=1,K)
      I = K + NCOMP
      CALL BCDBIN(ICSAVE,ISSAVE,3,$137)
       IF(NCOMP.GT.0.AND.I.NE.ISSAVE)GO TO 137
C
C              BUILD COMPUTED VARIABLES C008-CKKK IF MORE THAN 7 CXXX'S
C
C
C
      DO 130 I=8,K
      IF(K.LE.7)GO TO 130
      WRITE(28,902)I,I
 130  CONTINUE
C
C              WRITE OUT SOME DATA STATEMENTS, ETC USED BY GEN PROGRAM
C
C
C
      WRITE(28,921) ID1, ID2
      WRITE(28,1063)
      WRITE(28,1062)
      WRITE(28,970)
      WRITE(28,969)
      WRITE(28,947)
      WRITE(28,948)
      WRITE(28,1024)
      WRITE(28,1025)
      WRITE(28,1027)
      WRITE(28,949)
      WRITE(28,950)
      WRITE (28,979)
      WRITE (28,922) NCOMP
      WRITE(28,1078) NEWVAR
C
C
C              READ A CONTROL CARD
C
C
 108   READ(5,978) (ISAVE(I),I=1,14)
      IF(IPRT.EQ.0) CALL PRINTC
       DO 109 I=1,10
       IF(ISAVE(I).EQ.IOPRS(I)) GO TO 110
 109   CONTINUE
       I=8
C
C
C              GO TO APPROPRIATE ROUTINE
C
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

```
LINE #   TEXT...

 110    GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15),I
 C
 C
 136   WRITE(6,663)
 663   FORMAT(' OLD FORMAT TAPE: NO COMPUTED VARIABLE DICTIONARY.')
        GO TO 99999
 137    CONTINUE
        WRITE(6,1371) I,ICSAVE
 1371   FORMAT(I4,' COMPUTED VARIABLES EXPECTED,',A4,' READ')
        GO TO 99999
 138   WRITE(6,664)
 664   FORMAT(' DEFINITION CARD DOES NOT HAVE "C" OR "V" IN CC1.')
        GO TO 99999
 139   WRITE(6,666)
 666   FORMAT(' NEW BACKGROUND VARIABLES ARE MISS-NUMBERED.')
        GO TO 99999
 140    FORMAT(16H MISSING FORMAT ,A1,A3)
 141   WRITE(6,667) ISAVE(J),LCOMUN(J)
 667   FORMAT(' ONLY "T" OR "V" VARIABLES ARE ALLOWED ON STATEMENT',
      1 ' CARDS. ',/,' "',A1,J3,'" IS NOT VALID.')
        GO TO 99999
 11     CONTINUE
 12     CONTINUE
 13     CONTINUE
 14     CONTINUE
 15     CONTINUE
       WRITE(6,662) ISAVE(1), ISAVE(2)
 662   FORMAT(' NON-IMPLEMENTED CATAGORICAL OPERATOR = ',2A6)
        GO TO 99999
 C
 C
 C/OPEN
 C
 C
 1        WRITE(28,1057) ISAVE(1),ISAVE(2)
          WRITE(28,911)
          WRITE(28,923)
          WRITE(28,951)
          WRITE(28,924)
          WRITE(28,925)
          WRITE(28,926) NEWVAR
          WRITE(28,927)
          WRITE(28,928)
          WRITE(28,929)
          WRITE(28,930)
          WRITE(28,931)
          WRITE(28,932)
          WRITE(28,933)
 C
 C
```

```
LINE #   TEXT...
         C              WRITE RELATIVE LOCATION TABLE
         C
         C
                 IF(NEWVAR.EQ.0)GO TO 135
                 WRITE(28,1054) NEWVAR
                 WRITE(28,1055)
                 WRITE(28,1056)
                 WRITE(28,1012) NEWVAR
         135     WRITE(28,934)
                 WRITE(28,935)
                 WRITE(28,936)
                 WRITE(28,937)
         C
         C
         C              DICTIONARY CHANGES
         C
         C
                 IF(NVARS.EQ.0) GO TO 117
                 WRITE(28,987)
                 WRITE(28,988)
                 WRITE(28,989) NVARS
                 WRITE(28,990)
                 WRITE(28,991)
                 WRITE(28,992)
                 WRITE(28,1015)
                 WRITE(28,993)
                 WRITE(28,994)
                 WRITE(28,995)
         C
         C
         C              NEW COMPUTED VARIABLES
         C
         C
         117     IFMT = 9900
                 IF(NCOMP.EQ.0) GO TO 118
                  WRITE(28,938)
                  WRITE(28,939)
                  WRITE(28,940)
                  WRITE(28,941)
                  WRITE(28,942)
                  WRITE(28,943)
                  WRITE(28,944)
         118      I=0
                  WRITE(28,945)
                  N=NEWVAR+NVARS+NCOMP
                  WRITE(28,946)
         C
         C
         C              NEW BACKGROUND VARIABLES
         C
```

LINE # TEXT...

```
C
      IF(NEWVAR.EQ.0) GO TO 125
      WRITE(28,1009)NEWVAR
      WRITE(28,1010)
      WRITE(28,1011)
  125 WRITE(28,1075)
      WRITE(28,952)
      GO TO 108
C
C
C
C
C
C/READ
C
C
    2 WRITE(28,1057) ISAVE(1),ISAVE(2)
      J = 10
      WRITE(28,953) JREAD,J
       WRITE(28,1046)
      GO TO 108
C
C
C/DISPERSE
C
C
    3 WRITE(28,1057) ISAVE(1),ISAVE(2)
      DO 107 I=1,NTASK
      IF(NC.GT.0)GO TO 107
      J = 1
      WRITE(28,968) J
      WRITE(28,967) J,J
  107 CONTINUE
      WRITE(28,954)
      WRITE(28,955)
      K99998=ISAVE(3)
      IF(K99998.EQ.IBLNKS) K99998=K99997
      WRITE(28,997)       K99998
      WRITE(28,956)
      WRITE(28,957)
      WRITE(28,958)
      WRITE(28,959)
      WRITE(28,960)
      GO TO 108
C
C
C/COLLECT
C
C
    4 WRITE(28,1057) ISAVE(1),ISAVE(2)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      WRITE(28,998)
      WRITE(28,1031)
      WRITE(28,1029)
      WRITE(28,1030)
      WRITE(28,1047)
      WRITE(28,1032)
      WRITE(28,961)
      WRITE(28,1061)
      WRITE(28,962)
      WRITE(28,963)
      WRITE(28,964)
      WRITE(28,999)
      WRITE(28,1000)
      WRITE(28,965)
      WRITE(28,1059)
      WRITE(28,966)
      WRITE(28,1001) K99998
      WRITE(28,1002)
      WRITE(28,1003)
      WRITE(28,1033)
      GO TO 108
C
C
C/WRITE
C
C
5     WRITE(28,1057) ISAVE(1),ISAVE(2)
      WRITE(28,912)
      WRITE(28,1048)
      WRITE(28,1049)
      WRITE(28,1013) KRELOC
      J = 15
      WRITE(28,953) JWRIT,J
      GO TO 108
C
C
C/CLOSE
C
C
6     WRITE(28,1057) ISAVE(1),ISAVE(2)
      WRITE(28,977)
      WRITE(28,971)
      WRITE(28,972)
      WRITE(28,1071)
      WRITE(28,973)
      WRITE(28,1073)
      WRITE(28,1074)
      WRITE(28,1072)
      WRITE(28,974)
      WRITE(28,975)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          WRITE(28,976)
          WRITE(28,980)
         IFLAG = 1
          GO TO 108
C
C
C/END
C
C
7         WRITE(28,1057) ISAVE(1),ISAVE(2)
          IUNIT = 15
          IF(NCOMP.EQ.0.AND.NEWVAR.EQ.0.AND.NVARS.EQ.0) IUNIT = 10
          WRITE(28,1079) IUNIT
          WRITE(28,1080)
          WRITE(28,996)
          WRITE(28,981)
        ENDFILE 28
        CALL REWIND
        CALL SOOPN
500     CALL DREAD(NWORD,LOC,JSEQ,$500,$501)
        IF(NWORD.GT.14) NWORD = 14
        CALL SOPUT(LOC,NWORD)
        GO TO 500
501     CALL SOCLS
        CALL SPOST
         STOP
C
C
C          INSERT USER'S CARD IMAGES
C
C
8         WRITE(28,1077) (ISAVE(I),I=1,13),IUSER
          GO TO 108
C
C
C/AUDIT
C
C
9         WRITE(28,1057) ISAVE(1),ISAVE(2)
          WRITE(28,1076)
 111      GO TO 108
C
C
C/XQT
C
C
10        IF(IFLAG.NE.1) GO TO 162
          ENCODE(161,IOP)
161       FORMAT('@ADD Z*Z.XQT-PROGEN . ')
          CALL ERTRAN(6,IOP)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      GO TO 7
162    WRITE(6,163)
163    FORMAT(' BADLY PLACE "/XQT" CARD.  RUN HALTS.')
       CALL ERTRAN(2)
C
1100  WRITE(6,1101)
1101  FORMAT(' INVALID VARIABLE ID ABOVE')
99998 WRITE(6,668) ITYPE,NUM, JTYPE,INUM
668   FORMAT(1X,'NEW HISTORY & DICTIONARY CHANGES REQUIRE TWO CARDS.',
     1 /,1X,A1,A3,' WAS IMMEDIATELY FOLLOWED BY ',A1,A3)
99999 WRITE(6,99)
99    FORMAT(' RUN ABORTS DUE TO ABOVE ERROR.')
      CALL ERTRAN(2)
1026  CALL ERRMSG(4,'INVALID NUMERIC ABOVE')
      GO TO 99999
       END
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
.
.          DATA STORAGE
.
$(0)       AXR$
PARTBL.    RES        40                  . SIR/SOR PARAMETER TABLE
X11T       RES        1                   . X11 SAVE
ERM        'SISO INTERFACE ERROR AT RELATIVE LOCATION (A0)'
ERL        EQU        $-ERM
$(1)       ERU$
.
.          PRE-PROCESSOR ROUTINE
.                                    CALL 'SPREP' USING OPT-LIST, WB.
SPREP.     SX         X11,X11T
           LMJ        X11,PREPRO
           SLJ        ERR
           LXM        X11,X11T     .
           LXM        X11,0,X11    .
           LXI,U      X11,1        .
           LA         A2,(1,0)     .
           LA         A1,PARTBL    .
           LSSC       A1,10        .
           LR,U       R1,25        .
LOOP       SZ         A0           .
           LDSL       A0,1         .
           JZ         A0,ELOOP     .
           LA,U       A0,31        .
           ANA        A0,R1        .
           EX         STORIT,*A2   .
ELOOP      JGD        R1,LOOP      .
           LXM        X11,X11T     .
           J          2,X11        .
.
STORIT     SA,S1      A0,0,X11     .
           SA,S2      A0,0,X11     .
           SA,S3      A0,0,X11     .
           SA,S4      A0,0,X11     .
           SA,S5      A0,0,X11     .
           SA,S6      A0,0,X11     .
           LMJ        A3,$+1       .
           LXM,U      A2,1         .
           SA,S1      A0,1,*X11    .
           J          0,A3         .
.
.          POST PROCESSOR ROUTINE
.                                    CALL 'SPOST' USING WB.
SPOST.     SX         X11,X11T
           LMJ        X11,POSTPR
           SLJ        ERR
           J          XIT
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
.            SOURCE INPUT OPEN
.                                       CALL'SIOPN' USING WB.
SIOPN.       SX        X11,X11T
             LMJ       X11,OPNSRC
             SLJ       ERR
XIT          LXM       X11,X11T
             J         1,X11.
.
.            SOURCE INPUT CLOSE
.                                       CALL'SICLS' USING WB.
SICLS.       SX        X11,X11T
             LMJ       X11,CLOSRC
             SLJ       ERR
             J         XIT
.
.            SOURCE INPUT GET
.                                       CALL 'SIGET' USING WS-AREA, LEN, FLAG, WB.
SIGET.       SX        X11,X11T
SILP         LXM       A0,0,X11.
             LXI       A0,-1,X11.
             LMJ       X11,GETSRC
             SLJ       ERR
             J         EOF                 . EOF.. YES
             LXM       X11,X11T
             SA,H1     A3,-2,X11
             JP        A1,4,X11            . SOURCE CARD.. YES
             J         SILP                . NO,GET AGAIN
.
EOF          LXM       X11,X11T
             LXM       A0,1,X11
             SZ        0,A0
             J         4,X11
/
.            SOURCE OUTPUT OPEN
.                                       CALL 'SOOPN' USING WB.
SOOPN.       SX        X11,X11T
             LMJ       X11,SSOR
             SLJ       ERR
             J         XIT
.
.            SOURCE OUTPUT CLOSE
.                                       CALL 'SOCLS' USING WB.
SOCLS.       SX        X11,X11T
             LMJ       X11,ESOR
             SLJ       ERR
             J         XIT
.
.            SOURCE OUTPUT PUT
.                                       CALL 'SOPUT' USING LEN, AREA, WB.
SOPUT.       SX        X11,X11T
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
                LXM       A0,0,X11.
                LXI       A0,+1,X11.
                LMJ       X11,SOR
                SLJ       ERR
                LXM       X11,X11T
                J         3,X11
   /
   .            ERROR     ROUTINE
   .
   ERR          NOP
   PF           FORM      12,6,18
                L         A0,(PF 2,ERL,ERM)
                ER        PRINT$
                LX        X11,X11T
                LNA,U     A0,SIOPN
                AA,H2     A0,ERR
                ER        ERR$
   /.           LITERALS
   .
   .
                END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PRTVAR PRINT INFORMATION FROM A HISTORY/KPATH FILE W/SORT

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      C               PRINT KPATH ORDER WITH HISTORY AND COMPUTED VALUES
      C
      C
            DIMENSION LABEL(120),KA(2020),C(1), KC(50),FMT(48),KTAB(132),
           1 KB(2000),LTAB(1000),LHEAD(66),JPRT(132),ITYPE(100),NUM(100),
           2       ITC(8),KTST(2),IOPU(2),IH(6)
            EQUIVALENCE (FMT(1),KFMT), (KA,C)
                  KALF = 2**18
            IOPU(1)=6
            IOPU(2)=26
            KIN=10
            IJD = 27
            DATA IC/6HC     /,IV/6HV     /
            DATA KTST/12HEND OF DATA /
            CALL SETUP(10,27,26,0,0,0,0,21,0,0,0,JSEQ)
      C
        8   READ(5,899) LAB1,LAB2,IPTP
      899   FORMAT(7X,A6,A2,I2)
            KLINE=52
            IF(IPTP.EQ.0) IPTP = 2
                    REWIND KIN
                    READ(KIN) K,(KC(L), L =1,K)
            MATRIX = KC(6)
            NCASE = KC(3)
            IF(LAB1.EQ.KC(1).AND.LAB2.EQ.KC(2)) GO TO 9
            WRITE(6,898) KC(1),KC(2)
      898   FORMAT('1STUDY ID NO MATCH.  TAPE IS ',A6,A2,'.  RUN ABORTED.')
                CALL ERTRAN(2)
      C
        9           READ(KIN) KN,(LTAB(L),L =1,KN  )
                    K = KC(4)+1
      C  NEXT LINE WAS  NSKIP = K + 2 UNTIL CORRECTED ON 11 OCT 74
                NSKIP = K + 1
                  DO  10  L = 1,K
               READ(KIN) KM
       10         CONTINUE
             BACKSPACE KIN
       11    CONTINUE
                    LINE = KLINE +1
                    LPG   = 0
              JIM1 = 0
             READ(5,894,END=1000)KLAB,KHEAD,NVARS,NFMT,IS,IDSP,(ITC(I),I=1,8)
      894   FORMAT(I1,I2,I3,I2,1X,A1,1X,A6,1X,8A6)
            LSEQ = 1
            IF(IS.EQ.'1') LSEQ = 0
            IF(IS.EQ.'2') LSEQ = -1
            IDO2 = 1
            CALL BCDBIN(IS,IOP,1,$888)
            IF(IOP.LT.3) GO TO 888
            IF(IOP.GT.6) IOP = 3
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

```
LINE #   TEXT...

          IDO2 = 2
          MSEQ = 1
          IF(IOP.GE.5) LSEQ = 0
          IF(IOP.EQ.4.OR.IOP.EQ.6) MSEQ = 0
          GO TO 888
  888     CONTINUE
                   NLAB = KLAB*14
           IF(KLAB.GT.0) READ(5,895) (LABEL(L),L = 1,NLAB)
   895             FORMAT (13A6,A2)
                   NHEAD = KHEAD*22
           IF(NHEAD.GT.0) READ(5,896) (LHEAD(L),L= 1,NHEAD)
   896                    FORMAT (11A6)
          IF(NVARS.GT.0) GO TO 14
          WRITE(6,891)
   891     FORMAT ( 41H NO VARIABLE SPECIFICATIONS--RUN DELETED.)
          CALL ERTRAN(2)
   C
   C      THIS SECTION READS CODAP VARIABLES 16 PER CARD, COMPUTES THEIR
   C      RELATIVE LOCATION AND LENGTH AND STORES THAT INFO INTO KTAB.
   14     CONTINUE
          REWIND 21
          READ(5,15) (ITYPE(I),NUM(I),I=1,NVARS)
    15    FORMAT(16(A1,A3,1X))
          JTYPE = 0
          IF(ITYPE(1).EQ.'C') JTYPE = 1
          MTYPE = 0
          IF(ITYPE(2).EQ.'C') MTYPE = 1
          DO 30 I=1,NVARS
          CALL BCDBIN(NUM(I),INUM,3,$40)
          IF(ITYPE(I).EQ.IV) GO TO 20
          IF(ITYPE(I).NE.IC) GO TO 40
          IF(INUM.GT.KM) GO TO 40
          KTAB(I)=KALF+INUM
          GO TO 30
    20    IF(INUM.GT.KN) GO TO 40
          KTAB(I)=LTAB(INUM)+KM
    30    CONTINUE
          GO TO 91
    40    WRITE(6,50) ITYPE(I),NUM(I)
    50    FORMAT('1AN INVALID VARIABLE HAS BEEN ENCOUNTERED ON A CONTROL CAR
         1D.  IT IS ',A1,A3)
          CALL ERTRAN(2)
   C
   C
   C
    91             JFMT = NFMT*12
                   IF (NFMT.GT.0) GO TO 100
          WRITE(6,893)
            REWIND KIN
          CALL ERTRAN(2)
```

LINE #   TEXT...

```
 893          FORMAT(     40H NO FORMAT DATA PROVIDED,RUN TERMINATED. )
 100          READ(5,892) (FMT(L) , L= 1,JFMT )
  892         FORMAT (  12A6 )
       ISELCT = 1
       READ(5,800,END=1000) IEND,IHN,(IH(II),II=1,6),ISEEK
 800   FORMAT(A4,1X,A1,1X,6A6,33X,A4)
       KOUNT = 0
       IF(IEND.NE.'END') GO TO 112
       ISELCT = 0
       GO TO 110
 112   CONTINUE
       READ(5,800,END=1000) ISKIP
       DECODE(1776,IEND) IDS, ISUB
 1776  FORMAT(A1,A3)
       IF(IDS.NE.'C') GO TO 1775
       ISELCT = 2
       CALL BCDBIN(ISUB,ISUB,3,$997)
       IF(IHN.EQ.' ') IHN = '1'
       IF(IHN.EQ.'R') GO TO 1023
       CALL BCDBIN(IHN,IHN,1,$997)
       CEXL = IHN
       CEXH = IHN
       GO TO 110
 1023  CONTINUE
       DECODE(1024,IH(1)) CEXL
 1024  FORMAT(F6.0)
       DECODE(1024,IH(2)) CEXH
       GO TO 110
 1775  IF(IDS.NE.'V') GO TO 1777
       CALL BCDBIN(ISUB,ISUB,3,$997)
       CALL BCDBIN(IHN,IHN,1,$997)
       ISELCT = 3
       GO TO 110
 1777  CONTINUE
       CALL BCDBIN(ISEEK,ISEEK,4,$997)
       CALL GETMSK(ISEEK,LAB1,LAB2,MATRIX,$999)
 110   CONTINUE
       IF(ISELCT.EQ.3) ILOCP = MOD(LTAB(ISUB),KALF)
       DO 1100 ICASE = 1, NCASE
        READ(KIN) K,(KA(L),L=1,K),M,(KB(L),L=1,M)
           KK = K + M
           CALL TMTCSC(M+1,KA(K+1),KB(1))
           IF(ISELCT.EQ.0) GO TO 130
       IF(ISELCT.EQ.1) CALL USEMSK(ICASE,$1100)
       IF(ISELCT.NE.2) GO TO 1770
       KOUNT = KOUNT + 1
       IF(C(ISUB).LT.CEXL.OR.C(ISUB).GT.CEXH) GO TO 1100
       KOUNT = KOUNT - 1
       GO TO 130
 1770      IF(ISELCT.NE.3) GO TO 130
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      ILOCR = ILOCP + K
      DO 1771 II = 1, IHN
      IF(KA(ILOCR).EQ.IH(II)) GO TO 130
 1771 CONTINUE
      GO TO 1100
 130           KK = 1
      DO 150 KQ = 1,NVARS
               JL = MOD(KTAB(KQ),KALF)
               JK = JL + KTAB(KQ)/KALF-1
               DO 140  JM = JL,JK
                     JPRT(KK) = KA(JM)
                     KK = KK +1
  140          CONTINUE
  150       CONTINUE
          KK   = KK-1
      KLEN = KK
       JIM1 =JIM1 +1
      WRITE(21) (JPRT(L),L=1,KK)
 1100 CONTINUE
      ENDFILE 21
      REWIND 21
      NMEM = JIM1
      IF(NMEM.NE.0) GO TO 1778
      WRITE(6,1779)
 1779 FORMAT(' NO MEMBERS IN THIS GROUP.  REPORT IS BYPASSED.')
      IF(ISELCT.EQ.2) WRITE(6,1810) KOUNT, NCASE, ISUB, CEXL,CEXH
 1810 FORMAT(' REJECTED ',I5,' OUT OF ',I5,' CASES.  C',J3,
     1 ' WAS NOT BETWEEN ',F10.2,' AND ',F10.2)
      GO TO 200
 1778 CONTINUE
           IF(LSEQ.EQ.(-1)) GO TO 787
      IF(IDO2.EQ.1)  CALL FSORT(21,KLEN,JTYPE,LSEQ)
      IF(IDO2.EQ.2)  CALL FSORT2(21,KLEN,JTYPE,LSEQ,MTYPE,MSEQ)
 787       CONTINUE
      DO 1111 JIM1 = 1, NMEM
      READ(21) (JPRT(L),L=1,KLEN)
               LINE = LINE +1
               IF(LINE.LE.KLINE) GO TO 180
                LINE = 2
               LPG = LPG + 1
               DO 1003 JIM = 1,IPTP
               KOUT = IOPU(JIM)
      WRITE(KOUT,910) (ITC(I),I=1,8),IDSP,LPG
  910 FORMAT('1',8A6,29X,A6,' PAGE ',I3/)
          IF (NLAB.GT.0) WRITE(KOUT,911) (LABEL(L), L= 1,NLAB)
  911          FORMAT(1X,13A6,A2)
               WRITE (KOUT,911)
            IF (NHEAD.GT.0) WRITE (KOUT,912)  (LHEAD(L), L = 1,NHEAD)
  912                         FORMAT (22A6)
               WRITE (KOUT,911)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
                  IF (JIM.EQ.1)  LINE =LINE + KLAB + KHEAD
 1003  CONTINUE
 180   DO 1002 JIM = 1,IPTP
                  KOUT = IOPU(JIM)
                  WRITE (KOUT,KFMT) JIM1,(JPRT(L),L =1,KLEN)
 1002  CONTINUE
 1111  CONTINUE
       GO TO 200
 999   CONTINUE
       WRITE(6,998) ISEEK
 998   FORMAT('0ERROR ON JOB DESCRIPTION FILE SEARCHING FOR REPORT',
      1 ' FOR REPORT ',I6)
       GO TO 11
 1000  CONTINUE
 2000  CALL MARGIN
       STOP
 997   WRITE(6,996)
 996   FORMAT('0INVALID REPORT NUMBER IN REQUEST CARD.')
       READ(5,800,END=1000) ISKIP
       GO TO 11
 200   REWIND KIN
          DO 299 I = 1, NSKIP
          READ(KIN)
 299      CONTINUE
        IF(IPTP.EQ.2) CALL REPEND(LAB1,IDSP,ITC(1),JSEQ)
       REWIND 21
        GO TO 11
        END
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C         INTRODUCTION
C
C  PROGRAM FUNCTION
C
C     I. GENERAL - TO EVALUATE TASK RATINGS
C    II. SPECIFIC -
C        A. TO CALCULATE RATER CORRELATIONS WITH MEANS OF ALL RATERS
C        B. TO CALCULATE INTER-RATER RELIABILITY BASED ON RAW OR
C           ADJUSTED RATINGS
C        C. TO PRINT A DISTRIBUTION OF THE MEANS AND STANDARD DEVIATIONS
C           OF THOSE RATINGS
C
C  PROGRAM PROCEDURE
C
C     1. CHECK CONTROL CARDS, IF ERRORS SKIP TO 'END' AND RESTART
C     2. SCAN INPUT TAPE, U11
C        A. ACCUMULATE TASK SUMS
C        B. COMPUTE EACH RATER'S NTASK, MEAN, SD, RATINGS, AND COPY TO
C           DISK, U22
C     3. REWIND INPUT AND DISK
C     4. COMPUTE TASK MEANS
C     5. IF ADJUSTMENT IS REQUESTED, ZERO OUT THE TASK SUMS, ETC.
C     6. PASS DISK
C        A. PRINT CORRELATIONS UNLESS SUPPRESSED
C        B. RE-ACCUMULATE TASK DATA IF ADJUSTMENT IS SPECIFIED
C     7. COMPUTE MEANS, SD, SUMS OF SQUARES, AND DISTRIBUTIONS
C     8. CALCULATE INTER-RATER RELIABILITY
C     9. IF REQUESTED, PUNCH A MEANS DECK
C    10. PRINT RELIABILITY INFORMATION
C    11. PRINT THE DISTRIBUTIONS UNLESS SUPPRESSED
C    12. COPY TO BCD REPORT TAPE IF REQUIRED
C    13. RESET THE PROGRAM
C
C
      DIMENSION IHEAD(8), KARD(71), FMAT(71,50), ITLE(14,3), XR(1726),
     1NR(1726),LMN(81),LSD(81),TMN(1726),TSD(1726),TS(1726),TSQ(1726),
     2 TITMN(160), TITSD(160), IDATE(2), ITIME(2)
      EQUIVALENCE (TITMN(161),TITSD,TS(161)),(TS(1727),TSQ),
     1 (TSQ(1727),NR),(NR(1727),XR),(XR(1727),LMN,FMAT),
     2 (LMN(82),LSD),(LSD(82),TMN),(TMN(1727),TSD)
      DIMENSION IBASE(3), JBASE(3), KBASE(3), IDROP(1000), IHOLD(16)
      DIMENSION INFILE(1600), MBUF(14)
C
      DOUBLE PRECISION TRSQ,TRS,SQTK,STSK,TSUM,TSSQ,TSDTS,TSDSS,BSS
      INTEGER BLK, ONE
      INTEGER T, END, FMAT
      DATA KBASE, JBASE /18HRAW RATINGS       ,18HADJUSTED RATINGS
      DATA BLK, ONE, ICXXX, IZERO/ 1H , 1H1, 4HC..., '0'/
      DATA T, END /1HT, 3HEND/
      DATA IRS, IRR / 2HRS, 2HRR/
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

```
LINE #   TEXT...
          DATA (INFILE(I),I=1,11)/'10',6*' ',751,14,'IDSRRP',' '/
    C
          CALL SETUP(10,26,0,0,0,0,20,0,0,0,0,JSEQ)
          INP = 10
          IS = 20
          IPN = 1
          CALL RESET
          IRUNS = 0
          IGOOD = 0
    C
    C
    777   CALL ZERO(TS      ,3452)
          CALL ZERO(XR      ,1726 )
          CALL ZEROI(NR     ,1726 )
          REWIND IS
    C
    C
    C
    C     1. CHECK CONTROL CARDS, IF ERRORS SKIP TO 'END' AND RESTART
    C
    C  READ CONTROL CARD
          READ(5,800,END=999)NTIT,NCARD,IOP,JOP,KOP,IHEAD,IBCD,IDREPT,ISTUDY
          CALL PRINTC
          IRUNS = IRUNS + 1
    C
          CALL THTCSC(3,IBASE(1),KBASE(1))
          IF(IOP.NE.BLK) CALL THTCSC(3,IBASE(1),JBASE(1))
          KODE = IRR
          IF(IOP.EQ.ONE) KODE = IRS
          IF(IOP.EQ.IZERO) IOP = BLK
          IF(JOP.EQ.IZERO) JOP = BLK
    C
          IEND = BLK
          CALL BCDBIN(NCARD,NCARD,2,$902)
    C
          IF(NCARD.LE.0.OR.NCARD.GT.50) GO TO 902
    C
    C
          CALL BCDBIN(NTIT,NTIT,1,$901)
          IF(NTIT.GT.3) GO TO 901
          IF(NTIT.EQ.0) GO TO 2
    C
          DO 1 J = 1, NTIT
          READ(5,801) (ITLE(I,J),I=1,14)
          CALL PRINTC
    1     CONTINUE
    2     CONTINUE
    C
          READ(5,844) NDIGIT, IHI, LOW, IPRTSK, IDEL, SCALE
          CALL PRINTC
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      LOWF = 0
844   FORMAT(A1,1X,A6,1X,A6,1X,A1,1X,A1,1X,F6.0)
          IF(SCALE.LE.0.0) SCALE = 1.0
      IF(NDIGIT.EQ.' ') NDIGIT = '1'
      CALL BCDBIN(NDIGIT,NDIGIT,1,$906)
      IF(IHI.EQ.' ') IHI = '999999'
      IF(LOW.EQ.' ') LOWF = 1
      CALL BCDBIN(IHI,IHI,NDIGIT,$907)
      CALL BCDBIN(LOW,LOW,NDIGIT,$907)
      IF(LOWF.NE.0) LOW = LOWF
C
C READ ID'S OF CASES TO BE IGNORED.  STOP AFTER 'STOP'
      IDP = 0
      IF(IDEL.EQ.' ') GO TO 629
620   CONTINUE
      READ(5,621) IHOLD
      CALL PRINTC
621   FORMAT(16(A5))
      IF(IHOLD(1).EQ.'STOP') GO TO 629
      DO 622 I = 1, 16
      IF(IHOLD(I).EQ.' ') GO TO 622
      IDP = IDP + 1
      IF(IDP.GT.1000) GO TO 625
      IDROP(IDP) = IHOLD(I)
622   CONTINUE
      GO TO 620
625   CONTINUE
      CALL ERMMSG(8,'** NUMBER OF DELETED CASES EXCEEDS 1000.*****')
      GO TO 900
629   CONTINUE
C
C  SCAN FORMAT CARDS AND DETERMINE NTASK
      NTASK = 0
      DO 4 J = 1, NCARD
      READ(5,802) (FMAT(I,J),I=1,71)
      CALL PRINTC
      IF(FMAT(1,J).EQ.ICXXX) GO TO 3
C
      IEND = FMAT(1,J)
      GO TO 903
C
3     DO 4 I = 2, 70
4     IF(FMAT(I,J).EQ.T) NTASK = NTASK + 1
C
C PROCESS END CARD
      READ(5,802) IEND
      CALL PRINTC
      IF(IEND.NE.END) GO TO 902
C
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

REXALL INTER-RATER RELIABILITY W/ALL OPTIONS

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C        2. SCAN INPUT TAPE, UIO
C
         NCASE = 0
C
C
C  BEGIN CASE DATA LOOP
C
5          ITASK = 0
           JBAD = 0
           NTSK = 0
           STSK  = 0.0
           SQTK  = 0.0

C
C  ACCESS NEXT CASE
C
          DO 628 ICARD = 1, NCARD
          CALL READF(INFILE,MBUF,$10)
          DECODE(802,MBUF) KARD
          IF(ICARD.EQ.1) IDCASE = KARD(1)
          IF(IDCASE .EQ.END) GO TO 10
          IF(IDP.EQ.0) GO TO 626
          DO 627 I = 1, IDP
          IF(IDCASE.NE.IDROP(I)) GO TO 627
          JBAD = 1
          GO TO 628
   627    CONTINUE
   626    CONTINUE
          IF(IDCASE.EQ.KARD(1)) GO TO 6
C
          KARDN = ICARD
          ID = KARD(1)
          GO TO 904
C
6         IF(KARD(71).EQ.FMAT(71,ICARD)) GO TO 7
C
          KARDN = ICARD
          ICHKC = KARD(71)
          ICHKF = FMAT(71,ICARD)
          GO TO 905
C
7         DO 8 ICHAR = 2, 70
          IF(FMAT(ICHAR,ICARD).NE.T) GO TO 8
          ITASK = ITASK + 1
          LAST = ICHAR + NDIGIT - 1
          ENCODE(807,IVAL) (KARD(I),I=ICHAR,LAST)
          CALL BCDBIN(IVAL,IVAL,NDIGIT,$8)
          IF(IVAL.LT.LOW.OR.IVAL.GT.IHI) GO TO 8
          TSK = IVAL
               TSK = TSK/SCALE
          TSKSQ = TSK *  TSK
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
      NTSK = NTSK + 1
      STSK = STSK + TSK
      SQTK = SQTK + TSKSQ
      XR(ITASK) = TSK
C
C        A. ACCUMULATE TASK SUMS
C
      NR(ITASK) = NR(ITASK) + 1
      TS(ITASK) = TS(ITASK) + TSK
      TSQ(ITASK) = TSQ(ITASK) + TSKSQ
C
8     CONTINUE
628   CONTINUE
      IF(JBAD.EQ.1) GO TO 5
C
C        B. COMPUTE EACH RATER'S NTASK, MEAN, SD, RATINGS, AND COPY TO
C           DISK.
C
C
      NCASE = NCASE + 1
C
      X = NTSK
      XMN = STSK/X
      XSD = SQRT(ABS(SQTK/X - XMN * XMN))
C
C  WRITE RATER DATA TO DISK AND ACCESS NEXT RATER
      WRITE(IS) IDCASE, X, XMN, XSD, (XR(I),I=1,NTASK)
C
      CALL ZERO(XR     ,NTASK)
      GO TO 5
C  END OF LOOP FOR CASE DATA
C
C
C
C     3. REWIND DISK
C
10    CONTINUE
      REWIND IS
C
C
C
C     4. COMPUTE TASK MEANS
C
      NZTSK = 0
      NRESP = 0
      NRESQ = 0
      TXS = 0.0
      CALL ZERO(TMN,3452)
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

REXALL INTER-RATER RELIABILITY W/ALL OPTIONS

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      DO 11 ITASK = 1, NTASK
      IF(NR(ITASK).EQ.0) GO TO 11
      N = NR(ITASK)
      NRESP = NRESP + N
      NRESQ = NRESQ + N * N
      NZTSK = NZTSK + 1
C
      RN = N
      TMN(ITASK) = TS(ITASK) / RN
      TXS = TXS + TMN(ITASK)
C
11    CONTINUE
C
      Y = NTASK
      GMR = TXS / Y
      CX = 5.0 - GMR
C
C
C     5. IF ADJUSTMENT IS REQUESTED, ZERO OUT THE TASK SUMS, ETC.
C
      IF(IOP.NE.ONE) GO TO 12
C
      CALL ZERO(TS      ,3452)
C
C
C
C
12    IPAGE = 0
C
C   IF NEITHER THE STANDARDIZE NOR WRITE CORRELATIONS USED, GO TO 22
      IF(IOP.NE.ONE.AND.JOP.NE.BLK)  GO TO 22
C
C   ELSE ASSUME A WRITE CORRELATIONS BUT NO STANDARDIZING
      ASSIGN 20 TO IWRITE
      ASSIGN 16 TO IADJUS
C
C   AND CHANGE OPTIONS AS REQUIRED
      IF(IOP.EQ.ONE) ASSIGN 21 TO IADJUS
      IF(JOP.NE.BLK) ASSIGN 21 TO IWRITE
C
C     6. PASS DISK
C
      DO 16 ICASE = 1, NCASE
      DOT = 0.0
      STSK = 0.0
      SQTK = 0.0
C
C   SCAN DISK WITH RATER INFORMATION
      READ (IS) IDCASE, X, XMN, XSD, (XR(I),I=1,NTASK)
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
          DO 13 ITASK = 1, NTASK
          IF(XR(ITASK).EQ.0.0) GO TO 13
          Y  = TMN(ITASK)
          DOT = DOT + XR(ITASK) * Y
          STSK = STSK + Y
          SQTK = SQTK + Y * Y
   13     CONTINUE
C
          YGMN = STSK/X
C
C            A. PRINT CORRELATIONS UNLESS SUPRESSED
C
          GO TO IWRITE, (20, 21)
   20     CONTINUE
          YSD = SQRT(ABS(SQTK/X - YGMN * YGMN))
          CORR = (DOT/X - XMN * YGMN) / (XSD * YSD)
          CSQ = CORR * CORR
          IF(CSQ.LT.1.0) GO TO 23
          TV = 99999.0
          GO TO 26
   23     TV = 0.0
          IF(X.LT.2.0) GO TO 26
          TV = CORR * SQRT( X - 2.0) / SQRT(1.0 - CSQ)
   26     CONTINUE
C
          IF(MOD(ICASE,50).NE.1) GO TO 14
C
          IPAGE = IPAGE + 1
          WRITE(6,805) IPAGE
C
   14     WRITE(6,806) ICASE, CORR, X, XMN, XSD, YGMN, IDCASE, TV
C
C
C            B. RE-ACCUMULATE TASK DATA IF ADJUSTMENT IS SPECIFIED
C
          GO TO IADJUS, (21, 16)
   21     CONTINUE
C
          DO 15 ITASK = 1, NTASK
          IF(XR(ITASK).EQ.0.0) GO TO 15
          RS = (XR(ITASK) - XMN) / XSD + YGMN + CX
C
C   IF STANDARDIZATION WAS REQUESTED, RE-ACCUMULATE SUMS AND SUMS OF SQS
          TS(ITASK) = TS(ITASK) + RS
          TSQ(ITASK) = TSQ(ITASK) + RS * RS
C
   15     CONTINUE
   16     CONTINUE
C
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
22    CONTINUE
C
C
C       7. COMPUTE MEANS, SD, SUMS OF SQUARES, AND DISTRIBUTIONS
C
C  CALCULATE STATISTICS BASED ON DATA NOW STORED IN TS AND TSQ
C
        TRS = 0.0
        TRSQ = 0.0
        TSUM = 0.0
        TSSQ = 0.0
        TSDTS = 0.0
        TSDSS = 0.0
        BSS = 0.0
        CALL ZEROI(LMN,162)
        CALL ZERO(TMN     ,3452)
C
        DO 25 ITASK = 1, NTASK
        IF(NR(ITASK).EQ.0) GO TO 25
        RN = NR(ITASK)
C
C  ACCUMULATE DATA FOR TOTAL MEAN AND SD
        TRS = TRS + TS(ITASK)
        TRSQ = TRSQ + TSQ(ITASK)
C
C  DETERMINE MEAN AND SD FOR THIS TASK
        YMN = TS(ITASK) / RN
        YSD = SQRT(ABS(TSQ(ITASK)/RN - YMN * YMN))
        TMN(ITASK) = YMN
        TSD(ITASK) = YSD
C
C  ACCUMULATE DATA FOR MEAN OF MEANS
        TSUM = TSUM + YMN
C
C  ACCUMULATE DATA FOR SD OF MEANS
        TSSQ = TSSQ + YMN * YMN
C
C  ACCUMULATE DATA FOR MEAN OF SD
        TSDTS = TSDTS + YSD
C
C  ACCUMULATE DATA FOR SD OF SD
        TSDTSS = TSDTSS + YSD * YSD
C
C  CALCULATE BETWEEN SUM OF SQUARES (NOTE - TOTAL SUM OF SQS = TRSQ)
        BSS = BSS + (TS(ITASK)/RN) * TS(ITASK)
C
C
C  DISTRIBUTE THE MEAN AND SD FOR THIS TASK
C
        INTMN = (YMN - 1.0)/ 0.1  + 1.0
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
          IF(YMN.LT.1.0.OR.YMN.GT.9.0) INTMN = 81
          INTSD = (YSD - 0.0)/ .05  + 1.0
          IF(YSD.GT.4.0.OR.YSD.LT.0.0) INTSD = 81
          LMN(INTMN) = LMN(INTMN) + 1
          LSD(INTSD) = LSD(INTSD) + 1
C
25        CONTINUE
C
C
C   ESTABLISH DISTRIBUTION COLUMN HEADINGS IF REQUIRED
C
          IF(JOP.NE.BLK.AND.JOP.NE.ONE) GO TO 96
C
          CALL ZERO(TITMN  ,320 )
          A = 1.0
          B = 0.0
C
          DO 95 III = 1, 4
          I = III * 40 - 39
          J = III* 40 - 20
          DO 95 K = I, J
          L = K + 20
          TITMN(K) = A
          TITSD(K) = B
          A = A + 0.10
          B = B + 0.05
          TITMN(L) = A - 0.01
95        TITSD(L) = B - 0.01
          TITMN(160) = 9.00
          TITSD(160) = 4.00
C
96        CONTINUE
C
C
C       8. CALCULATE INTER-RATER RELIABILITY
C
          W = NRESP
          W2 = NRESQ
          X = NCASE
          Y = NTASK
          Z = NZTSK
C
C   COMPUTE CORRECTION FACTOR, C
          C =(TRS/W) * TRS
C
C   COMPUTE THE SUMS OF SQUARES
C
          TSS = TRSQ - C
          BSS = BSS - C
          WSS = TSS - BSS
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C
C
C  COMPUTE XK, THE AVERAGE NUMBER OF RATINGS PER TASK COEFFICIENT
C
       XTASK = NZTSK - 1
       XK = (W - W2/W) / XTASK
C
C
C  COMPUTE THE DEGRESS OF FREEDOM
C
       DFB = XTASK
       DFW = NRESP - NCASE - NZTSK + 1
       IF(IOP.NE.ONE) DFW = NRESP - NZTSK
       DFT = DFB + DFW
C
C
C   COMPUTE BETWEEN TASK AND WITHIN MEAN SQUARES
C
       BTMS = BSS / DFB
       WMS = WSS / DFW
C
C
C  COMPUTE RATER RELIABILITY FOR A SINGLE RATER  (R11)
C
       R11 = (BTMS - WMS)/(BTMS + (XK - 1.0) * WMS)
C
C  AS A NOTE, TO COMPUTE RYY, SIMPLY DIVIDE XK BY Y. FOR THIS REASON,
C      RKK (WHERE K = XK)  =  (BTMS - WMS) / BTMS
C
C  COMPUTE THE INTER-RATER RELIABILITY FOR THIS SAMPLE OF K RATERS
       RKK = (BTMS - WMS) / BTMS
C
C  AN EQUIVALENT FORM FOR RYY, GIVEN R11 IS
C           RYY =  (Y*R11) /(1. + (Y - 1.)* R11)
C
C
        F = BTMS / WMS
C
C
C  COMPUTE TOTAL TASK MEAN AND SD
       TSKMNR = TRS / W
       TSKSDR = SQRT(ABS(TRSQ/W - TSKMNR * TSKMNR))
C
C  COMPUTE MEAN TASK MEAN AND SD
       TSKMNA = TSUM / Z
       TSKSDA = TSDTS / Z
C
C  COMPUTE SD OF TASK MEAN AND SD
       SDMN = SQRT(ABS(TSSQ/Z - TSKMNA * TSKMNA))
       SDSD = SQRT(ABS(TSDTSS/Z - TSKSDA * TSKSDA))
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      C
      C
      C     9. IF REQUESTED, PUNCH A MEANS DECK
      C
            IF(KOP.NE.ONE) GO TO 93
      C
            IF(IOP.NE.ONE) GO TO 710
      C
      C  IF ADJUSTMENT REQUIRED, RE-CALCULATE MEANS, DISTRIBUTION, + STATS
      C
           TSUM = 0.0
           TSSQ = 0.0
           CALL ZEROI(LMN    ,80  )
           DO 700 ITASK = 1, NTASK
           IF(TMN(ITASK).EQ.0.0) GO TO 700
           YMN = (TMN(ITASK) - TSKMNA) / SDMN + 5.0
           TMN(ITASK) = YMN
           TSUM = TSUM + YMN
           TSSQ = TSSQ + YMN * YMN
           INTMN = (YMN - 1.0) / 0.1 + 1.0
           IF(YMN.LT.1.0.OR.YMN.GT.9.0) INTMN = 81
           LMN(INTMN) = LMN(INTMN) + 1
  700      CONTINUE
           TSKMNA = TSUM / Z
           SDMN = SQRT(ABS(TSSQ / Z - TSKMNA * TSKMNA))
  710      CONTINUE
      C
           RANHI = (10 ** NDIGIT) / SCALE
           JFORM = 1
           IF(RANHI.GT.10.1) JFORM = 2
           CALL DATETM(IDATE,ITIME)
           WRITE(IPN,4811) IDREPT, IHEAD, IDATE, ITIME, IBASE(1)
 4811      FORMAT(A6,1X,8A6,1X,A6,A3,1X,A6,A2,3X,A3)
           GO TO (644,645), JFORM
  644      CONTINUE
           LX = (NTASK + 11) / 12
           NVALS = 12
           WRITE(IPN,5811) NTASK, LX, NVALS
 5811      FORMAT('NTASK = ',J4,' NCARDS = ',J3,' NVALS/CARD = ',J2)
           IF(LX.LE.99) WRITE(IPN,1811)
           IF(LX.GT.99) WRITE(IPN,3811)
           DO 646 III = 1, LX
           J = III * 12 - 11
           K = III * 12
           IF(K.GT.NTASK) K = NTASK
           IF(LX.LE.99) WRITE(IPN,1815) III,KODE,ISTUDY,(TMN(I),I=J,K)
           IF(LX.GT.99) WRITE(IPN,3815) III,KODE,ISTUDY,(TMN(I),I=J,K)
  646      CONTINUE
           GO TO 93
  645      CONTINUE
```

LINE #   TEXT...

```
          LX = (NTASK + 6) / 7
          NVALS = 7
          WRITE(IPN,5811) NTASK, LX, NVALS
          IF(LX.LE.99) WRITE(IPN,811)
          IF(LX.GT.99) WRITE(IPN,2811)
          DO 647 III = 1, LX
          J = III * 7 - 6
          K = III * 7
          IF(K.GT.NTASK) K = NTASK
          IF(LX.LE.99) WRITE(IPN,815) III,KODE,ISTUDY,(TMN(I),I=J,K)
          IF(LX.GT.99) WRITE(IPN,2815) III,KODE,ISTUDY,(TMN(I),I=J,K)
  647     CONTINUE
C
  93      CONTINUE
C
C
          PCTRSP = W/(X * Z) * 100.0
          IOUT = 6
  101     CONTINUE
C
C       10. PRINT RELIABILITY INFORMATION
C
          WRITE(IOUT,830) IHEAD, IDREPT
          IF(NTIT.LE.0) GO TO 102
          WRITE(IOUT,831) ((ITLE(I,J),I=1,14),J=1,NTIT)
          WRITE(IOUT,888)
  102     WRITE(IOUT,832) IBASE
          WRITE(IOUT,856) R11
          WRITE(IOUT,857) RKK
          WRITE(IOUT,858) F
          WRITE(IOUT,860) DFB
          WRITE(IOUT,861) DFW
          WRITE(IOUT,859) DFT
          WRITE(IOUT,864) C
          WRITE(IOUT,866) BSS
          WRITE(IOUT,867) WSS
          WRITE(IOUT,865) TSS
          WRITE(IOUT,862) BTMS
          WRITE(IOUT,863) WMS
          WRITE(IOUT,853) XK
          WRITE(IOUT,852) X
          WRITE(IOUT,850) Y
          WRITE(IOUT,851) Z
          WRITE(IOUT,854) W
          WRITE(IOUT,855) PCTRSP
          WRITE(IOUT,868) TSKMNR
          WRITE(IOUT,869) TSKSDR
          WRITE(IOUT,870) TSKMNA
          WRITE(IOUT,871) TSKSDA
          WRITE(IOUT,872) SDMN
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

```
LINE #   TEXT...
            WRITE(IOUT,873) SDSD
      C
      C     11. PRINT THE DISTRIBUTIONS UNLESS SUPRESSED
      C
            IF(JOP.NE.BLK.AND.JOP.NE.ONE) GO TO 200
      C
            IREMAN = NZTSK - LMN(81)
            WRITE(IOUT,840) IHEAD, IDREPT, IBASE
            WRITE(IOUT,841) W,TSKMNR,TSKSDR,Z,TSKMNA,SDMN,X,TSKSDA,SDSD,
           1 IREMAN, LMN(81)
            DO 150 III = 1, 4
            I = III * 40 - 39
            J = III* 40
            K = III * 20 - 19
            L = III * 20
      150   WRITE(IOUT,842) (TITMN(M),M=I,J), (LMN(N),N=K,L)
            IREMAN = NZTSK - LSD(81)
            WRITE(IOUT,843) IREMAN, LSD(81)
            DO 175 III = 1, 4
            I = III * 40 - 39
            J = III* 40
            K = III * 20 - 19
            L = III * 20
      175   WRITE(IOUT,842) (TITSD(M),M=I,J), (LSD(N),N=K,L)
      C
      C     12. COPY TO BCD REPORT TAPE IF REQUIRED
      C
      200   IF(IBCD.EQ.ONE) GO TO 9997
            IF(IOUT.EQ.26) GO TO 250
            IOUT = 26
      C
            GO TO 101
      C
      250   CALL REPEND(ISTUDY,IDREPT,IHEAD,JSEQ)
      C
      C
      C     13. PRINT TASK SUMMARY IF REQUESTED, AND RESET THE PROGRAM
      C
      9997  CONTINUE
            IF(IPRTSK.EQ.' ') GO TO 849
            INDEX = 0
      848   CONTINUE
            WRITE(6,845) IBASE
      845   FORMAT('1TASK MEANS AND STANDARD DEVIATIONS IN DUTY-TASK',
           1 ' ORDER BASED ON ',3A6,
           2 ///,2(1X,'  TASK   RATERS       MEAN       S.D.',9X),
           3 1X,'  TASK   RATERS       MEAN       S.D.',/)
            DO 847 JTASK = 1, 50
            I = JTASK + INDEX
            IF(I.GT.NTASK) GO TO 849
```

LINE # TEXT...

```
      J = JTASK + INDEX + 50
      K = JTASK + INDEX + 100
      IFORM = 3
      IF(K.GT.NTASK) IFORM = 2
      IF(J.GT.NTASK) IFORM = 1
      GO TO (641,642,643), IFORM
641   WRITE(6,846) I,NR(I),TMN(I),TSD(I)
      GO TO 847
642   WRITE(6,846) I,NR(I),TMN(I),TSD(I),J,NR(J),TMN(J),TSD(J)
      GO TO 847
643   WRITE(6,846) I,NR(I),TMN(I),TSD(I),J,NR(J),TMN(J),TSD(J),K,NR(K),
     1 TMN(K),TSD(K)
847   CONTINUE
      INDEX = INDEX + 150
      IF(I.GE.NTASK.OR.J.GE.NTASK.OR.K.GE.NTASK) GO TO 849
      GO TO 848
849   CONTINUE
846   FORMAT(2(1X,I6,3X,I6,2F10.3,9X),1X,I6,3X,I6,2F10.3)
      CALL RESET
      CALL CLOSEF(INFILE)
      INFILE(9) = 14
      CALL ZEROI(INFILE(12),63)
      IGOOD = IGOOD + 1
      GO TO 777
C
800   FORMAT(7X,A1,1X,A2,1X,A1,1X,A1,1X,A1,1X,8A6,1X,A1,1X,A6,1X,A4)
801   FORMAT(13A6,A2)
802   FORMAT(A4,69A1,A6)
803   FORMAT(7X,'CARD ',I2,' OF CASE ID ',A4,' HAD ID = ',A4)
804   FORMAT(7X,'FORMAT CARD',I3,' HAD CHECK CODE = ',A6,'. THAT CARD IN
     1 CASE ',A4,' WAS = ',A6)
805   FORMAT( '1RATER CORRELATIONS WITH SAMPLE USING RAW RESPONSES',15X,
     112X,20X,I4,//,'   ID    CORRELATION    NTASK    MEAN      S.D.   S
     2AMPLE MEAN',5X,'CASE ID',5X,'T-VALUE',//)
806   FORMAT(1X,I4,5X,F10.5,F9.0,F8.2,F10.4,4X,F10.2,8X,A4,F12.3)
807   FORMAT(6A1)
811   FORMAT('(I2,A2,A4,7F10.3)')
1811  FORMAT('(I2,A2,A4,12F6.3)')
2811  FORMAT('(I3,A1,A4,7F10.3)')
3811  FORMAT('(I3,A1,A4,12F6.3)')
815   FORMAT(I2,A2,  A4,7F10.3)
1815  FORMAT(I2,A2,A4,12F6.3)
2815  FORMAT(I3,A1,A4,7F10.3)
3815  FORMAT(I3,A1,A4,12F6.3)
C
830   FORMAT('1',8A6,29X,A6,1X,'PAGE     1',/)
831   FORMAT(1X,13A5,A2)
832   FORMAT(15X,'INTER-RATER RELIABILITY BASED ON ',3A6,/)
C
840   FORMAT('1',8A6,29X,A6,1X,'PAGE     2',//,15X,'DISTRIBUTION OF MEANS
```

```
LINE #   TEXT...

         1 AND STANDARD DEVIATIONS BASED ON ',3A6,/)
  841     FORMAT(7X,'NRESP  = ',F8.0,14X,'MEAN OF ALL RESPONSES = ',F10.4,
         114X,'S.D. OF ALL RESPONSES = ',F10.4,/,7X,'NZTSK  = ',F8.0,14X,
         2'MEAN OF MEAN RATINGS  = ',F10.4,14X,'S.D. OF MEAN RATINGS  = ',
         3F10.4,/,7X,'NRATER = ',F8.0,14X,'MEAN OF ALL THE S.D.S = ',F10.4,
         414X,'S.D. OF ALL THE S.D.S = ',F10.4,///,20X,'DISTRIBUTION OF MEAN
         5S',19X,I6,' IN RANGE',5X,I6,' OUT OF RANGE')
  842     FORMAT(1H0,/,1H ,4X,20F6.2,/,5X,20F6.2,/,5X,20I6)
  843     FORMAT(//,20X,'DISTRIBUTION OF STANDARD DEVIATIONS',5X,I6,' IN',
         1 ' RANGE',5X,I6,' OUT OF RANGE')
C
  850     FORMAT('0NTASK  = ',F10.0,2X,'NUMBER OF TASKS TO BE RATED         ')
  851     FORMAT('0NZTASK = ',F10.0,2X,'NUMBER OF TASKS ACTUALLY RATED      ')
  852     FORMAT('0NRATER = ',F10.0,2X,'TOTAL NUMBER OF RATERS              ')
  853     FORMAT('0K      = ',F10.2,2X,'AVERAGE NUMBER OF RATERS PER TASK   ')
  854     FORMAT('0NRESP  = ',F10.0,2X,'TOTAL NUMBER OF RATINGS             ')
  855     FORMAT('0PCTRSP = ',F10.2,2X,'PERCENT OF TOTAL POSSIBLE RATINGS   ')
  856     FORMAT('0R11    = ',F10.5,2X,'RELIABILITY FOR A SINGLE RATER      ')
  857     FORMAT('0RKK    = ',F10.5,2X,'RELIABILITY FOR THESE K RATERS      ')
  858     FORMAT('0F11    = ',F10.4,2X,'F-RATIO (BTMS/WMS)                  ')
  859     FORMAT('0DFT    = ',F10.0,2X,'DEGREES OF FREEDOM  (TOTAL)         ')
C
  860     FORMAT('0DFB    = ',F10.0,2X,'DEGREES OF FREEDOM  (BETWEEN, N1)   ')
  861     FORMAT('0DFW    = ',F10.0,2X,'DEGREES OF FREEDOM  (WITHIN, N2)    ')
  862     FORMAT('0BTMS   = ',F10.2,2X,'BETWEEN TASKS MEAN SQUARES          ')
  863     FORMAT('0WMS    = ',F10.2,2X,'WITHIN TASKS MEAN SQUARES           ')
  864     FORMAT('0C      = ',F10.0,2X,'CORRECTION FACTOR FOR SUM OF SQS    ')
  865     FORMAT('0TSS    = ',F10.0,2X,'TOTAL SUM OF SQUARES                ')
  866     FORMAT('0BSS    = ',F10.0,2X,'BETWEEN SUM OF SQUARES              ')
  867     FORMAT('0WSS    = ',F10.0,2X,'WITHIN SUM OF SQUARES               ')
  868     FORMAT('0TOTMN  = ',F10.4,2X,'MEAN OF ALL TASK RATINGS            ')
  869     FORMAT('0TOTSD  = ',F10.4,2X,'STANDARD DEVIATION OF ALL RATINGS   ')
C
  870     FORMAT('0AVEMN  = ',F10.4,2X,'MEAN OF ALL TASK MEANS              ')
  871     FORMAT('0AVESD  = ',F10.4,2X,'MEAN OF TASK STANDARD DEVIATIONS    ')
  872     FORMAT('0SDMN   = ',F10.4,2X,'STANDARD DEVIATION OF TASK MEANS    ')
  873     FORMAT('0SDSD   = ',F10.4,2X,'STANDARD DEVIATION OF STANDARD DEVIA
         1TIONS')
C
  888     FORMAT('0')
C
  900     CALL ERRMSG(7,'** SKIPPING TO END CARD DUE TO ABOVE ERROR')
  9000    READ(5,801) IEND
          CALL PRINTC
          IF(IEND.NE.END) GO TO 9000
C
  9001    CALL RESET
          CALL CLOSEF(INFILE)
          INFILE(9) = 14
          CALL ZEROI(INFILE(12),83)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      GO TO 777
C
901   CALL ERRMSG(8,'** NUMBER OF TITLE CARDS IS NOT IN THE RANGE 0-3')
      GO TO 900
C
902   CALL ERRMSG(7,'** NUMBER OF FORMAT CARDS IS NOT CORRECT  ')
      GO TO 900
C
903   CALL ERRMSG(8,'** C... IS NOT IN CC 1-4 OF THIS FORMAT CARD    ')
      IF(IEND.NE.END) GO TO 900
      GO TO 9001
C
904   CALL ERRMSG(6,'** CASE ID MISMATCH. RUN ABORTS    ')
      WRITE(6,803) KARDN, IDCASE, ID
      GO TO 9001
C
905   CALL ERRMSG(7,'** CHECK CODE DOES NOT MATCH. RUN ABORTS  ')
      WRITE(6,804) KARDN, ICHKF, IDCASE, ICHKC
      GO TO 9001
C
C
906   CALL ERRMSG(7,'** INVALID NUMBER OF DIGITS SPECIFIED.     ')
      GO TO 900
907   CALL ERRMSG(7,'** HI & LOW MUST CONTAIN NDIGITS, LEFT ADJ.')
      GO TO 900
C
999   IF(IEND.EQ.END) GO TO 1000
      CALL ERRMSG(7,'UNEXPECTED END OF FILE ON CARD READER     ')
1000  CALL ERRMSG(3,'** END OF REXALL  ')
      STOP
       END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C                              - DECK LOGIC -
C     1. INITIALIZE PROGRAM AND CHECK CONTRO. CARD VALUES (THRU STATEMENT
C     31).
C     2. GENERATE FORMAT AND COMPARE WITH CONTROL CARD PARAMETERS (THRU
C     300).
C     3. MOVE CASE BY GENERAL FORMAT FROM INPUT TO SCRATCH, CHECKING NUM-
C     BER OF CARDS (TO 315).
C     4. MOVE CASE BY GENERATED FORMAT FROM SCRATCH TO OUTPUT, CHECKING
C     TASK VALUES AND CHECK CHARACTERS (TO 901).
C     5. ERROR ROUTINES, CLOSEOUTS, AND FORMATS (TO END).
C
      COMMON/FGEN/IFMT(1000),ITSKTB(1700),IHISTB(1700),ICHKTB(500),
     1ICHEK(500),LFWTB(50),NTHTB(50),NCHEK,NCARD,KTH,KTT
      DIMENSION ICCARD(12),INCARD(14),IVCARD(14),LODFMT(52),INFO(3850),
     1KARD(14)
      INTEGER SCUNIT,OPUNIT
      DIMENSION INUNIT(1600), OPUNIT(1600), IVUNIT(1600), MBUF(14)
      DATA IZERO/'0'/
      DATA NBLANK/6H      /, NZERO/6H000000/
      DATA (INUNIT(I),I=1,11)/'10',6*' ',751,14,'ISDRLP',' '/
      DATA (OPUNIT(I),I=1,11)/'15',6*' ',751,14,'OSDRRP',' '/
      DATA (IVUNIT(I),I=1,11)/'16',6*' ',751,14,'OSDRRP',' '/
      NZER = 0
      NONE = 1
      NTWO = 2
      NFOR = 4
      NFIV = 5
      NSIX = 6
      NOTSK = 1
      IADDM1 = 1
      ISKIP1 = 0
      IFLAG1 = 0
      KUMUWD = 0
      INVALK = 0
      INCR9 = 0
      DO 5 INDEX = 1,3850
      INFO(INDEX) = 0
    5 CONTINUE
      ASSIGN 302 TO LCF
      ASSIGN 722 TO JBY
C  SPECIFY MAX. STORAGE FOR THE  HISTORY VARIABLES.
      MAXSTO = 1700
C  SPECIFY I/O UNITS.
      CALL SETUP(10,15,16,0,0,0,20,0,0,0,0,JSEQ)
      SCUNIT = 20
C
C  REWIND I/O UNITS
      REWIND SCUNIT
C
      CALL LINCNT
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

SETCHK CHECK CASE DATA SETUP

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C
C
C  READ CONTROL CARD.
      READ(5,801,END=901) (ICCARD(I),I=1,12)
      MAXTSK = ICCARD(2)
      MAXSGL = ICCARD(3)
      NCARD = ICCARD(5)
      NTASK = ICCARD(7)
      NHIST = ICCARD(9)
      NDUTY = ICCARD(11)
C  PERFORM ERROR CHECKS 10 - 14.
      CALL BCDBIN(MAXTSK,MAXTSK,NFIV,$907)
      MAXTSK = MAXTSK * 10
      IF(MAXTSK .NE. 0) GO TO 7
      ASSIGN 341 TO IGUIDE
      GO TO 8
 7    CONTINUE
      ASSIGN 340 TO IGUIDE
 8    CONTINUE
      CALL BCDBIN(MAXSGL,MAXADD,NONE,$907)
      MAXTSK = MAXTSK + MAXADD
      IF(MAXTSK.LT.1) GO TO 907
 10   CONTINUE
      CALL BCDBIN(NCARD,NCARD,NTWO,$908)
      IF(NCARD.LT.1.OR.NCARD.GT.50) GO TO 908
 11   CONTINUE
      IF(NTASK.EQ.' ') GO TO 909
      CALL BCDBIN(NTASK,NTASK,NFOR,$909)
      IF(NTASK.LT.0.OR.NTASK.GT.2000) GO TO 909
 12   CONTINUE
      CALL BCDBIN(NHIST,NHIST,NFOR,$910)
      IF(NHIST.LT.0.OR.NHIST.GT.1000) GO TO 910
 13   CONTINUE
      CALL BCDBIN(NDUTY,NDUTY,NTWO,$911)
      IF(NDUTY.LT.0.OR.NDUTY.GT.26) GO TO 911
 25   CONTINUE
C  TEST ERROR FLAG.
      IF(IFLAG1.EQ.0) GO TO 29
C  PRINT CONTROL CARD AND ABORT.
      WRITE(6,802) (ICCARD(I),I = 1,12)
      CALL ERTRAN(2)
 29   CONTINUE
C  DETERMINE 'NUMBER OF PLACES' IN MAXTSK.
      INCR2 = 0
      MSF = MAXTSK
 30   CONTINUE
      INCR2 = INCR2 + 1
      MSF = MSF/10
      IF(MSF.EQ.0) GO TO 31
      GO TO 30
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 31     CONTINUE
 C  GENERATE FORMAT.
        CALL FMTGEN(NTWO)
        IF(NCHEK.EQ.0) ASSIGN 332 TO JBY
 C  PERFORM ERROR CHECKS 1 - 3.
        IF(KTH.EQ.NHIST) GO TO 32
        WRITE(6,804) KTH,NHIST
        CALL ERTRAN(2)
 32     CONTINUE
        IF(KTT.EQ.NTASK) GO TO 34
        WRITE(6,805) KTT,NTASK
        CALL ERTRAN(2)
 34     CONTINUE
        DO 35 J = 1,KTH
        KALWRD = IHISTB(J)
 C  CONVERT NUMBER OF CHAR. TO NUMBER OF WORDS.
        KALWRD = (KALWRD/64**3 -1)/6 +1
        KUMUWD = KUMUWD + KALWRD
 35     CONTINUE
        IF(KUMUWD.LE.MAXSTO) GO TO 300
        WRITE(6,806) KUMUWD,MAXSTO
        CALL ERTRAN(2)
 C
 C
 C
 C  CASE COMPILATION LEVEL
 C
 C
 300    CONTINUE
 C  SET END-OF-CASE-FILE EXIT.
 C  READ A CARD OF DATA BY GENERAL FORMAT.
        ICARD = 0
 301    CONTINUE
        CALL READF(INUNIT,INCARD,$912)
        DECODE(808,INCARD(1)) IDPROB
        IF(ICARD.NE.0) GO TO 303
 C  SET CARD I.D. COMPARITOR.
 302    CONTINUE
        IDTEST = IDPROB
 303    CONTINUE
        IF(IDPROB.NE.IDTEST) GO TO 305
 C  WRITE CARD TO SCRATCH UNIT.
        WRITE(SCUNIT,810) (INCARD(J),J = 1,14)
        ICARD = ICARD + 1
        GO TO 301
 305    CONTINUE
 C  END OF CASE
        REWIND SCUNIT
 C  TEST FOR PROPER NUMBER OF CARDS IN CASE.
        IF(ICARD - NCARD) 310,315,307
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 307   CONTINUE
       CALL LINCNT
 C  PRINT ERROR MESSAGE 6.
       WRITE(6,811) IDTEST,ICARD
       GO TO 311
 310   CONTINUE
       CALL LINCNT
 C  PRINT ERROR MESSAGE 5.
       WRITE(6,812) IDTEST,ICARD
 C
 C  ROUTINE TO COPY ONTO INVALID TAPE FILE.
 311   CONTINUE
       DO 314 INDEXO = 1,ICARD
       READ(SCUNIT,810) (IVCARD(K),K = 1,14)
       GO TO (313,312),NOTSK
 312   CONTINUE
       IF(INDEXO.NE.INCR1) GO TO 313
       CALL LINCNT
       CALL LINCNT
       WRITE(6,813) ITCHEK,KREF,(IVCARD(K),K = 1,14)
       NOTSK = 1
 313   CONTINUE
       CALL WRITEF(IVUNIT,IVCARD)
 314   CONTINUE
       INVALK = INVALK + 1
       REWIND SCUNIT
 C  CHECK LAST-CASE FLAG AND PROCESS NEXT CASE IF AVAILABLE.
       ICARD = 0
       GO TO LCF,(302,915)
 C
 C
 C  CARD VERIFICATION LEVEL
 C
 C
 315   CONTINUE
 C  LODFMT ROUTINE
 C  GENERATE ARRAY BLOC BOUNDARY SUBSCRIPTS.
       LASTWD = 0
       KLOSWD = 0
       DO 320 INDEX1 = 1,NCARD
       IRSTWD = LASTWD + 1
       LASTWD = LFWTB(INDEX1)
       INTRWD = KLOSWD + 1
       KLOSWD = NTHTB(INDEX1)
 C  EXTRACT GENERATED FORMAT.
       INCRD = 0
       DO 316 INDEX2 = IRSTWD,LASTWD
       INCRD = INCRD + 1
       LODFMT(INCRD) = IFMT(INDEX2)
 316   CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
         GO TO (317,318),NOTSK
  317    CONTINUE
C  READ CARD INTO CORE BY GENERATED FORMAT
         READ(SCUNIT,LODFMT) (INFO(INDEX3),INDEX3 = INTRWD,KLOSWD)
         GO TO 320
  318    CONTINUE
C  WRITE EDITED CARD ONTO VALID CASE TAPE BY GENERATED FORMAT.
         ENCODE(LODFMT,MBUF) (INFO(INDEX9),INDEX9=INTRWD,KLOSWD)
         CALL WRITEF(OPUNIT,MBUF)
  320    CONTINUE
         GO TO (322,321),NOTSK
  321    CONTINUE
         NOTSK = 1
         ICARD = 0
         GO TO LCF,(302,915)
  322    CONTINUE
         REWIND SCUNIT
C
C  VERIFY CHECK CHARACTERS.
         GO TO JBY,(722,332)
  722    CONTINUE
         DO 325 INDEX4 = 1,NCHEK
         LOCVAL = ICHKTB(INDEX4)
         ITCHEK = INFO(LOCVAL)
         IF(ITCHEK.EQ.ICHEK(INDEX4)) GO TO 325
C  NO MATCH - FIND CURRENT CARD BOUNDARIES.
         KREF = ICHEK(INDEX4)
         DO 323 INDEX5 = 1,NCARD
         INCR1 = INDEX5
         IF(LOCVAL.LE.NTHTB(INDEX5)) GO TO 327
  323    CONTINUE
  325    CONTINUE
         GO TO 332
C  PRINT ERROR MESSAGE 4 WHILE COPYING CASE TO INVALID FILE.
  327    CONTINUE
         ICARD = NCARD
         NOTSK = 2
         GO TO 311
  332    CONTINUE
C
C  VERIFY TASK RANGES ON EACH CARD.
C  REPLACE OUT-OF-RANGE VALUES AND/OR BLANKS WITH ZEROES.
         INIT = 1
         KRDNUM = 1
         KRDOOR = 0
         IFLAG2 = 0
         IRFLAG = 0
         IF(NTASK.EQ.0) IRFLAG = 1
         ASSIGN 348 TO IVEER
         DO 350 INDEX7 = 1,KTT
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      LOCT = ITSKTB(INDEX7)
      KSAT = INFO(LOCT)
C  TEST FOR BLANK.
      IF(KSAT.EQ.NBLANK) GO TO 343
C  TEST TASK RANGE.
      GO TO IGUIDE,(341,340)
340   CONTINUE
      CALL BCDBIN(KSAT,KSAT,INCR2,$342)
      IF(KSAT.EQ.NZER) GO TO 344
      IF(KSAT.GT.NZER.AND.KSAT.LE.MAXTSK) GC TO 744
      GO TO 342
341   CONTINUE
      IF(KSAT.EQ.IZERO) GO TO 344
      IF(KSAT.GT.IZERO.AND.KSAT.LE.MAXSGL) GO TO 744
342   CONTINUE
C  ZERO ROUTINE SEGMENT
      KRDOOR = KRDOOR + 1
      IFLAG2 = 1
343   CONTINUE
C  ZEROING STATION
      INFO(LOCT) = NZERO
      GO TO 344
744   CONTINUE
C  IF AT LEAST ONE NONZERO TASK IS IN RANGE, SET FLAG.
      IRFLAG = 1
344   CONTINUE
C  TEST FOR LAST TASK.
      IF (INDEX7 .EQ. KTT) GO TO 349
      MORE1 = INDEX7 + 1
      LOCTP1 = ITSKTB(MORE1)
345   CONTINUE
C  TEST FOR CHANGE OF CARD.
      IF (LOCTP1 .LE. NTHTB(KRDNUM)) GO TO 350
346   CONTINUE
C  TEST TASK-OUT-OF-RANGE FLAG.
      IF (IFLAG2 .EQ. 0) GO TO IVEER,(348,350)
      IFLAG2 = 0
C  FIND CURRENT CARD ON SCRATCH UNIT.
      DO 347 INDEX8 = INIT,KRDNUM
      READ(SCUNIT,810) (KARD(L),L = 1,14)
347   CONTINUE
      INIT = KRDNUM + 1
      CALL LINCNT
      CALL LINCNT
C  PRINT ERROR MESSAGE 7.
      WRITE(6,814) KRDOOR,(KARD(L),L = 1,14)
      KRDOOR = 0
      GO TO IVEER,(348,350)
348   CONTINUE
C  PROCESS NEXT CARD
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      KRDNUM = KRDNUM + 1
      GO TO 345
349   CONTINUE
      ASSIGN 350 TO IVEER
      GO TO 346
350   CONTINUE
      REWIND SCUNIT
      IF(IRFLAG.EQ.1) GO TO 355
      CALL LINCNT
C  PRINT ERROR MESSAGE 8.
      WRITE(6,815) INFO(1)
      GO TO 311
355   CONTINUE
C  INCREMENT VALID CASE COUNTER.
      INCR9 = INCR9 + 1
      NOTSK = 2
      GO TO 315
C
C
C
C
C
C
C  ERROR ROUTINES
901   CONTINUE
      WRITE(6,866)
      CALL ERTRAN(2)
907   CONTINUE
      IFLAG1 = 1
      WRITE(6,867)
      GO TO 10
908   CONTINUE
      IFLAG1 = 1
      WRITE(6,868)
      GO TO 11
909   CONTINUE
      IFLAG1 = 1
      WRITE(6,869)
      GO TO 12
910   CONTINUE
      IFLAG1 = 1
      WRITE(6,870)
      GO TO 13
911   CONTINUE
      IFLAG1 = 1
      WRITE(6,871)
      GO TO 25
C
C  END OF FILE ON DATA TAPE.
C  SET LAST-CASE FLAG AND PROCESS LAST CASE.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

```
LINE #   TEXT...

  912   CONTINUE
        ASSIGN 915 TO LCF
        GO TO 305
  915   CONTINUE
        CALL CLOSEF(IVUNIT)
        CALL CLOSEF(OPUNIT)
  C  PROCESSING COMPLETE - PRINT CLOSEOUT MESSAGES.
        CASTOT = INVALK + INCR9
        BAD = INVALK * 100
        PCD = BAD / CASTOT
        WRITE(6,880) INCR9,INVALK,PCD,INCR9,NTASK,NDUTY,NHIST
        STOP
  C
  801   FORMAT(A6,A5,A1,A1,A2,A1,A4,A1,A4,A1,A2,A6)
  802   FORMAT(1H0,10X,'CONTROL CARD',/6X,A6,A5,A1,A1,A2,A1,A4,A1,A4,A1,A2
       1,A6)
  804   FORMAT(1H0,5X,I4,' HISTORY VARIABLES WERE DEFINED BY FORMAT CARDS
       1AND ',I4,' HISTORY VARIABLES WERE',/,6X,'INDICATED ON THE CONTROL
       2CARD. RUN ABORTED.')
  805   FORMAT(1H0,5X,I4,' TASK VARIABLES WERE DEFINED BY FORMAT CARDS AND
       1 ',I4,' TASK VARIABLES WERE',/,6X,'INDICATED ON THE CONTROL CARD.
       2RUN ABORTED.')
  806   FORMAT(1H0,5X,I5,' COMPUTER WORDS ARE REQUIRED TO STORE HISTORY VA
       1RIABLES.',/6X,'THIS EXCEEDS THE MAXIMUM OF ',I5,'. RUN ABORTED.')
  808   FORMAT(A4)
  810   FORMAT(14A6)
  811   FORMAT(1H0,5X,'TOO MANY CARDS IN  CASE ',A4,' .  ',I6,' CARDS FOUN
       1D. CASE DELETED.')
  812   FORMAT(1H0,5X,'TOO FEW CARDS IN CASE ',A4,' .  ONLY ',I2,' CARD(S)
       1 FOUND. CASE DELETED.')
  813   FORMAT(1H0,5X,'CHECK CHARACTER NO MATCH. CASE DELETED.',/,6X,'CASE
       1 CHECK CHARACTER IS ',A2,'  AND CONTROL CHECK CHARACTER IS ',A2,'.
       2  CARD IMAGE IS',/6X,14A6)
  814   FORMAT(1H0,5X,I2,' OUT OF RANGE TASK VALUE(S) SET TO ZERO. CARD IM
       1AGE IS ',/6X,14A6)
  815   FORMAT(1H0,5X,'CASE ',A4,' HAD NO TASKS WITHIN RANGE. CASE DELETED
       1.')
  866   FORMAT(1H0,5X,'UNEXPECTED END OF FILE IN CONTROL/FORMAT DECK. RUN
       1ABORTED.')
  867   FORMAT(1H0,5X,'THE MAXIMUM ACCEPTABLE VALUE FOR A TASK, SPECIFIED
       1BY CONTROL CARD, IS INCORRECT.',/,6X,'RUN ABORTED.')
  868   FORMAT(1H0,5X,'THE NUMBER OF CARDS PER CASE SPECIFIED BY CONTROL C
       1ARD IS INCORRECT. RUN ABORTED.')
  869   FORMAT(1H0,5X,'THE NUMBER OF TASKS SPECIFIED BY CONTROL CARD IS IN
       1CORRECT. RUN ABORTED.')
  870   FORMAT(1H0,5X,'THE NUMBER OF HISTORY VARIABLES SPECIFIED BY CONTRO
       1L CARD IS INCORRECT. RUN ABORTED.')
  871   FORMAT(1H0,5X,'THE NUMBER OF DUTIES SPECIFIED BY CONTROL CARD IS I
       1NCORRECT. RUN ABORTED.')
  880   FORMAT(1H1,5X,'SETCHK COMPLETE.',//,5X,I6,2X,'CASES ACCEPTED',/,5X
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      1,I6,2X,'CASES DELETED',/,5X,F6.1,2X,'PERCENT CASES DELETED',///,6X
      2,'EDITED RAW DATA TAPE CONTAINS -',//,5X,I6,2X,'CASES',/,5X,I6,2X,
      3'TASKS',/,5X,I6,2X,'DUTIES',/,5X,I6,2X,'HISTORY VARIABLES')
       SUBROUTINE LINCNT
       DATA IPAGE , LINES /1,100/
       LINES = LINES + 2
       IF(LINES.LE.56) RETURN
       WRITE(6,800) IPAGE
       IPAGE = IPAGE + 1
       LINES = 2
       RETURN
  800  FORMAT('1',46X,'ERROR MESSAGES',46X,'PAGE',I3,/)
       END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

TSETUP SETUP JOB DESCRIPTION AND/OR REPORT FILES

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      DIMENSION IMAGE(8), NAME1(2), NAME2(2), IFREE(3)
      DATA IBLANK/' '/
      READ(5,100) IDPROG, NAME1, NAME2,IBIAS
      CALL BCDBIN(IBIAS,JBIAS,4,$900)
      NUMONE = JBIAS + 1
      CALL ERTRAN(9,IDATE,ITIME)
      WRITE(6,101) IDPROG, IDATE, ITIME
      ENCODE(107,IFREE)
      WRITE(6,105) IFREE
      CALL FREE(15)
      IF(NAME1(1).EQ.IBLANK) GO TO 10
      IF(NAME1(1).EQ.'SCRATC') GO TO 10
      DO 5 ICYCLE = 1, 3
      ENCODE(102,IMAGE) NAME1
      WRITE(6,105) (IMAGE(I),I=1,6)
          CALL GETFIL(IMAGE)
      ENCODE(103,IMAGE) NAME1
      WRITE(6,105) (IMAGE(I),I=1,5)
      CALL ERTRAN(6,IMAGE)
      REWIND 15
      ENDFILE 15
      REWIND 15
          CALL FREES(15)
      WRITE(6,188) ICYCLE
      CALL DELAY(10)
  188 FORMAT(' REEL ',I1,' OF 3 REELS IS INITIALIZED',/)
    5 CONTINUE
   10 CONTINUE
      IF(NAME2(1).EQ.IBLANK) GO TO 20
      IF(NAME2(1).EQ.'SCRATC') GO TO 20
      WRITE(6,155) NUMONE
  155 FORMAT(' REPORT FILE BEGINS WITH SENTINEL = ',I4)
      DO 15 ICYCLE = 1, 3
      ENCODE(102,IMAGE) NAME2
      WRITE(6,105) (IMAGE(I),I=1,6)
          CALL GETFIL(IMAGE)
      ENCODE(103,IMAGE) NAME2
      WRITE(6,105) (IMAGE(I),I=1,5)
      CALL ERTRAN(6,IMAGE)
      REWIND 15
      WRITE(15,106) NUMONE
      ENDFILE 15
      REWIND 15
          CALL FREES(15)
      WRITE(6,188) ICYCLE
      CALL DELAY(10)
   15 CONTINUE
   20 CONTINUE
      STOP
  900 CONTINUE
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      WRITE(6,156) IBIAS
156   FORMAT(' AN INVALID "LAST" SENTINEL VALUE = "',A4,'".')
      CALL ERTRAN(2)
100   FORMAT(A6,1X,A6,A4,2X,A6,A4,2X,A4)
101   FORMAT('1DYNAMICALLY ISSUED CONTROL STATEMENTS FOR ',A6,56X,A6,
     1 1X,A6,//)
102   FORMAT('@ASG,U  ',A6,A4,'M(+1).,16N . .      ')
103   FORMAT('@USE 15,',A6,A4,'M(+1). . ')
105   FORMAT(1X,8A6)
106   FORMAT('BEGIN REPORT',A6)
107   FORMAT('@FREE,A 15. . ')
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      COMMON KE(500),KD(1726),TMEAN(1726),SD(1726),MOPT,ACASE,
     1 IFMTWK(3452),IDATWK(3452),ISID1
      DIMENSION INFMT(71),INDAT(71),SUMX(1726),SUMX2(1726),ACNT(1726),
     1 ADATWK(3452),KFMT(12)
      DIMENSION AY(6600)
      EQUIVALENCE (ADATWK,IDATWK),(TMEAN,SUMX),(SD,SUMX2),
     1 (INFMT,INDAT),(ACNT,KD)
     DIMENSION INFILE(1600), MBUF(14), ILAB(8)
     DATA (INFILE(I),I=1,11)/'11',6*' ',751,14,'IDSRRP',' '/
      DATA ITEE/6HT     /,IBLANK/6H      /
      CALL SETUP(10,11,26,0,0,0,20,21,0,0,0,JSEQ)
      NAVAIL=6600
      ICON=2**30
      INP=5
     ASSIGN 600 TO INPUT
      DO 5 I=1,3452
      IFMTWK(I)=0
      IDATWK(I)=0
    5 CONTINUE
      DO 6 I=1,1726
      SUMX(I)=0.
      SUMX2(I)=0.
      ACNT(I)=0.
    6 CONTINUE
C
C     READ I.D. CARD
      READ 1, ISID1,ISID2,MOPT,IPNCH,NCARD,MIN,MAX,NCASE
    1 FORMAT(7X,A6,A2,I2,I2,I3,X,A1,I2,I5)
     IF(MAX.EQ.0) MAX = 7
      AMAX=MAX
      IF(MIN.EQ.IBLANK) GO TO 180
      CALL BCDBIN(MIN,MIN,1,$200)
      GO TO 190
  180 MIN=1
  190 AMIN=MIN
C
C     READ COMMUNICATION REGION
      IHT=10
      REWIND IHT
      READ(IHT) K,(KE(I),I=1,K)
      IF(ISID1.EQ.KE(1).AND.ISID2.EQ.KE(2)) GO TO 4
      WRITE (6,14)
   14 FORMAT (38H1STUDY I.D. NO MATCH.  RUN TERMINATED.)
      GO TO 120
    4 NTASK=KE(12)
      ACASE=KE(3)
      IMAX=NAVAIL/(NTASK*3+75)
    9 IF(MOPT.LT.1.OR.MOPT.GT.3) GO TO 89
      GO TO (11,15,12),MOPT
   11 INP=11
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
            ASSIGN 610 TO INPUT
            GO TO 15
     12     READ 87,(KFMT(I),I=1,12)
     87     FORMAT(12A6)
            READ(0,88) NVALS
      88    FORMAT(10X,I2)
            LNUM=0
            IPT=1
      90    KPT=IPT+NVALS-1
            IF(KPT.GT.NTASK) KPT=NTASK
            READ KFMT,KNUM,X,X,(TMEAN(I),I=IPT,KPT)
            IF(KNUM.LE.LNUM) GO TO 91
            LNUM=KNUM
            IPT=KPT+1
            IF(KPT.NE.NTASK) GO TO 90
            GO TO 112
      91    PRINT 95
      95    FORMAT('1WEIGHT CARDS OUT OF SEQUENCE.  RUN ABORTED')
            STOP
     200    ICOL = 24
            GO TO 201
      89    ICOL = 17
     201    WRITE(6,32) ICOL
      32    FORMAT('1COLUMN ',I2,' OF I.D. CARD IN ERROR.  RUN ABORTED')
            GO TO 120
       15   NCOL=NCARD*71
            J=1
 C
 C          READ AND STACK FORMAT CARDS
            DO 10 I=1,NCARD
            READ 2, (INFMT(N),N=1,71)
        2   FORMAT(A4,69A1,I2)
            DO 20 K=1,71
            IFMTWK(J)=INFMT(K)
            J=J+1
      20    CONTINUE
      10    CONTINUE
 C
 C          READ AND STACK RATER DATA
      16    DO 100 IX=1,NCASE
            J=1
           GO TO INPUT
 600       CONTINUE
            DO 30 I=1,NCARD
            READ(INP,3)(INDAT(N),N=1,71)
        3   FORMAT(A4,69A1,I2)
            DO 40 K=1,71
            IDATWK(J)=INDAT(K)
            J=J+1
      40    CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
   30 CONTINUE
      GO TO 650
  610 DO 630 I = 1, NCARD
      CALL READF(INFILE,MBUF,$650)
      DECODE(3,MBUF) (INDAT(N),N=1,71)
      DO 640 K = 1, 71
      IDATWK(J) = INDAT(K)
      J = J + 1
  640 CONTINUE
  630 CONTINUE
  650 CONTINUE
       K=1
       ICASE=IDATWK(1)
       DO 50 I=.,NCARD
C
C      CHECK CASE CONTROL AND CARD NUMBER
       IF(IDATWK(K).EQ.ICASE.AND.IDATWK(K+70).EQ.IFMTWK(K+70)) GO TO 45
       GO TO 51
   45  K=K+71
   50  CONTINUE
       K=0
       DO 60 I=1,NCOL
C
C      IF FORMAT COLUMN IS T CHECK DATA COLUMN FOR VALID RESPONSE
       IF(IFMTWK(I).NE.ITEE) GO TO 60
       K=K+1
       CALL BCDBIN(IDATWK(I),IDATWK(I),1,$56)
       ADATWK(I)=IDATWK(I)
       IF(ADATWK(I).LT.AMIN.OR.ADATWK(I).GT.AMAX) GO TO 55
       SUMX(K)=SUMX(K)+ADATWK(I)
       SUMX2(K)=SUMX2(K)+ADATWK(I)**2
       ACNT(K)=ACNT(K)+1.
       GO TO 60
   55  IF(ADATWK(I).EQ.0.) GO TO 60
   56  WRITE(6,7) IDATWK(I),K
    7  FORMAT(' CASE ',A4,' HAS RATING ',I3,' OUT OF RANGE.  RATING SKIP
      1PED.')
   60  CONTINUE
       IF(K.EQ.NTASK)GO TO 100
          IDIF = K - NTASK
       WRITE(6,46) K,NTASK,IDIF
   46  FORMAT(' THE NUMBER OF T''S IN FORMAT DOES NOT MATCH NUMBER F',
      1'ROM COMM REGION.',/,1X,I6,' T''S PRESENT',/,1X,I6,' = NTASK',/,
      1 1X,I6,' = DIFFERENCE',/,' RUN TERMINATED.')
       GO TO 120
  100  CONTINUE
C
C      COMPUTE MEANS, SD
       DO 110 I=1,NTASK
       SD(I)=SQRT((SUMX2(I)*ACNT(I)-SUMX(I)**2))/ACNT(I)
```

LINE # TEXT...

```
          TMEAN(I)=SUMX(I)/ACNT(I)
  110   CONTINUE
C
C       PUNCH MEANS IF REQUESTED
        KPUN = ' '
        IF(IPNCH.EQ.2.OR.IPNCH.EQ.3) KPUN = 'X'
        IF(IPNCH.EQ.2) IPNCH = 0
        IF(IPNCH.EQ.0) GO TO 112
        WRITE(1,42)
   42   FORMAT('(I2,A2,A4,12F6.2)')
        ICARD=1
        IPT=1
  111   KPT=IPT+11
        IF(KPT.GT.NTASK)KPT=NTASK
        WRITE(1,44) ICARD,ISID1,(TMEAN(I),I=IPT,KPT)
   44   FORMAT(I2,'TX',A4,12F6.2)
        ICARD=ICARD+1
        IPT=KPT+1
        IF(KPT.NE.NTASK) GO TO 111
        GO TO 112
   51   WRITE(6,52)IDATWK(K)
   52   FORMAT(15H1ERROR IN CASE ,A4,28H OF RATER DECK.  RUN ABORTED)
        GO TO 120
C
C       SET UP STORAGE FOR VARIABLE DIMENSIONS
  112   N1 = 9*IMAX+1
        N2 = N1+36*IMAX
        N3 = N2+8*IMAX
        N4 = N3+9*IMAX
        N5 = N4+IMAX
        N6 = N5+NTASK*IMAX
        N7 = N6+NTASK*IMAX
        N8 = N7+IMAX
        N9 = N8+IMAX
        N10 = N9+IMAX
        N11 = N10+IMAX
        LGH=N11+IMAX
        CALL TSKNL1 (IMAX,NTASK,AY,AY(N1),AY(N2),AY(N3),AY(N4),AY(N5),
     1 AY(N6),AY(N7),AY(N8),AY(N9),AY(N10),AY(LGH),JSEQ)
        ENDFILE 21
        REWIND 21
        WRITE(6,1200)
 1200   FORMAT('1SUMMARY OF TSKGRP RUN:',/,109X,'TASKS',
     1/,' SEQ  IDREPT  TITLE',45X,'CUM % TIME   AVE % MEM   AVE TASK',
     2 ' DIF  NON-ZERO  TOTAL  MEMBERS')
        LSEQ = 0
 1201   READ(21,END=1220) K,ID,ILAB,CUM,AVE,DIF,TOTX,KNT,MEM
        N = TOTX
        LSEQ = LSEQ + 1
        WRITE(6,1202) LSEQ,ID,ILAB,CUM,AVE,DIF,N,KNT,MEM
```

COMPUTATICNAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
1202  FORMAT(1X,I3,2X,A6,2X,8A6,2X  7.2,6X,F7.2,8X,F5.2,7X,I4,5X,I4,
     1 4X,I4)
      IF(KPUN.NE.' ') WRITE(1,1203)ID,(ILAB(I),I=1,6),CUM,AVE,DIF,N,
     1 KNT,MEM
1203  FORMAT(A6,1X,6A6,F7.2,F7.2,F5.2,3(I4))
      GO TO 1201
1220  WRITE(6,130)
120   STOP
  130  FORMAT('1')
       SUBROUTINE TSKNL1 (IMAX,NTASK,IVL,IVAR,IOPN,ISGT,NNN,ACNT,SUMS,
     1 IDSP,IBAD,IBCD,NMEM,ITGRP,JSEQ)
       COMMON KE(500),KD(1726),TMEAN(1726),SD(1726),MOPT,ACASE,
     1 IFMTWK(3452),IDATWK(3452),ISID1
       DIMENSION IVL(9,IMAX),IVAR(36,IMAX),IOPN(8,IMAX),ISGT(9,IMAX),
     1 NNN(IMAX),ACNT(NTASK,IMAX),IBCD(IMAX),IDSP(IMAX),IBAD(IMAX),
     2 SUMS(NTASK,IMAX),NMEM(IMAX),ITGRP(NTASK,IMAX),ICARD(20)
       DIMENSION IREL(1726),HEAD(14,6),IBUF(2,2),KTST(9),KOUNT(20),
     1 INUM(5),  KB(1726),ILAB(8),FBUF(2,2),ITSK(1726),IMVBUF(1726),
     2 INX(1726),IOUT(2),LBUF(54),MMIN(1726)
       EQUIVALENCE (IBUF,FBUF),        (INX(1),IFMTWK(1)),
     1 (IMVBUF(1),KB(1),IFMTWK(1727)),          (IREL(1),ITSK(1)),
     2 IDATWK(1)),(MMIN(1),IDATWK(1727))
       DATA IEND/6HEND   /,KEND/6HEND OF/
      IPASS = 0
       IMT=10
       ISCR=20
      ISC1 = 21
      REWIND ISC1
       INUM(1)=2**30
       INUM(2)=2**24
       INUM(3)=2**18
       INUM(4)=2**12
       INUM(5)=2**6
       IOUT(1)=6
       IOUT(2)=26
       KTASK=NTASK-1
       NDICT=KE(4)
C
C      READ RELATIVE LOCATION TABLE.
 107   READ(IMT) K,(IREL(I),I=1,K)
       IUSED=0
       REWIND ISCR
C
C      SKIP DICTIONARY.
       DO 110 I=1,NDICT
 110   READ(IMT) K
C
C      READ AND CHECK CONTROL CARD.
       NJD=1
  112  READ(5,3,END=113) JEND,ISEQ,IHEAD,IBCD(NJD),IPRNT,IDSP(NJD),
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
     1 (ILAB(I),I=1,8)
   3  FORMAT(A3,I1,3I2,1X,A6,1X,8A6)
      IF(MOPT.EQ.3) IPRNT=1
      IF(IBCD(NJD).EQ.0) IBCD(NJD)=2
      IF(ISEQ.GE.1.AND.ISEQ.LE.5) GO TO 115
      WRITE(6,15)IDSP(NJD)
  15  FORMAT(29H1INCORRECT SORT SEQUENCE FOR ,A6,13H.  ASSUMED 1.)
      ISEQ=1
      GO TO 115
 113  LEND=IEND
      IF(NJD.GT.1) GO TO 141
      GO TO 407
 115  IF(IHEAD.GE.0.AND.IHEAD.LE.6) GO TO 120
      WRITE(6,17)IDSP(NJD)
  17  FORMAT(41H1THE NUMBER OF TITLE CARDS SPECIFIED FOR ,A6,27H IN ERR
     1OR.  RUN TERMINATED.)
      GO TO 407
 120  IF(IPRNT.EQ.1.OR.IPRNT.EQ.2) GO TO 121
      WRITE(6,19) IDSP(NJD)
  19  FORMAT(40H1PRINT OPTION INCORRECTLY SPECIFIED FOR ,A6,20H.  ASSUM
     1ED OPTION 2.)
      IPRNT = 2
 121  IF(IBCD(NJD).GE.1.AND.IBCD(NJD).LE.2) GO TO 125
      WRITE(6,21)IDSP(NJD)
  21  FORMAT(43H1BCD TAPE OPTION INCORRECTLY SPECIFIED FOR ,A6,13H.  AS
     1SUMED 1.)
      IBCD(NJD)=1
C
125         CONTINUE
      CALL ZERO(ITGRP(1,NJD),NTASK)
      KOUNT(NJD) = 0
      IXBAD = 0
1119  READ(5,1120) ICARD
1120  FORMAT(20A4)
      DO 1124 I = 1, 20
      IF(ICARD(I).EQ.'STOP') GO TO 1125
      IF(ICARD(I).EQ.'    ') GO TO 1124
      CALL BCDBIN(ICARD(I),MTASK,4,$1123)
      IF(MTASK.GT.NTASK.OR.MTASK.LT.1) WRITE(6,1222) MTASK
      IF(MTASK.GT.NTASK.OR.MTASK.LT.1) GO TO 1123
1222  FORMAT(' SPECIFIED TASK OUT OF RANGE = ',I6)
      ITGRP(MTASK,NJD) = 1
      KOUNT(NJD) = KOUNT(NJD) + 1
      GO TO 1124
1123  CONTINUE
      IXBAD = IXBAD + 1
      WRITE(6,1122) IDSP(NJD),ICARD(I)
1122  FORMAT(1X,A6,' BAD TASK SPECIFIED = *',A4,'*, IGNORED')
      IF(IXBAD.LT.9) GO TO 1124
      WRITE(6,1121)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 1121 FORMAT(' RUN ABORTS DUE TO EXCESSIVE BAD TASKS.')
      CALL ERTRAN(2)
 1124 CONTINUE
      GO TO 1119
C
C      READ TITLE CARDS.
 1125 CONTINUE
      IF(IHEAD.GT.0) READ(5,5)((HEAD(I,K),I=1,14),K=1,IHEAD)
      IF(IHEAD.EQ.0) CALL BLANK(14,HEAD)
      IF(IHEAD.EQ.0) IHEAD = 1
    5 FORMAT(13A6,A2)
C
C      READ VARIABLE NAMES AND OPERATORS
       JERR=0
      IPRNTS = 0
      CALL SAMPLE(NJD,ISGT(1,NJD),NN,JERR,IPRNTS)
      NNN(NJD) = NN
       N=1
       DO 146 K=1,NN
       IF(ISGT(K,NJD).GE.4) GO TO 135
       IF(ISGT(K,NJD).GE.0) GO TO 130
C
C      READ LIMITS - ALPHANUMERIC
       READ 9,L,KTST(K),((IBUF(J,M),M=1,L),J=1,2)
    9  FORMAT(I2,2X,I3,1X,4A6)
       KTST(K)=-KTST(K)
       GO TO 140
C
C      READ LIMITS - INTEGER
  130  READ 11,L,KTST(K),IBUF(1,1),IBUF(2,1)
   11  FORMAT(I2,2X,I3,1X,I6,6X,I6)
       GO TO 140
C
C      READ LIMITS - FLOATING POINT
  135  READ 13,KTST(K),FBUF(1,1),FBUF(2,1)
   13  FORMAT(4X,I3,1X,2F12.6)
       L = 1
 140   KL = L*2
       IF(ISGT(K,NJD).NE.KTST(K))JERR=JERR+1
C
C      TEST TO SEE IF ANY ERRORS IN VARIABLE ACTION CARDS
       IF(JERR.EQ.0) GO TO 142
       WRITE(6,23)IDSP(NJD)
   23  FORMAT(54H1ERROR IN VARIABLE INTERACTION OR DEFINITION CARD FOR ,
     1A6,'.  RUN ABORTED.')
           CALL ERTRAN(2)
  142  DO 145 M=1,L
       J=N+M-1
       IVAR(J,NJD)=IBUF(1,M)
       J=J+L
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
  145 IVAR(J,NJD)=IBUF(2,M)
      N=N+KL
      IVL(K,NJD)=L
  146 CONTINUE
C
C     WRITE CONTROL,INDEX TITLE, AND TITLE CARDS
      WRITE(ISCR,3) JEND,ISEQ,IHEAD,IBCD(NJD),IPRNT,IDSP(NJD),
     1 (ILAB(I),I=1,8)
      WRITE(ISCR,5)((HEAD(I,K),I=1,14),K=1,IHEAD)
C
C     TEST IF MAX NUMBER OF GROUPS HAS BEEN REACHED.
      IF(NJD.GE.IMAX) GO TO 147
      NJD=NJD+1
      GO TO 112
C
  141 NJD=NJD-1
  147 CONTINUE
      IPASS = IPASS + 1
C
C     CLEAR TABLES
 157  REWIND ISCR
      DO 151 J=1,NJD
      DO 150 I=1,NTASK
      SUMS(I,J)=0.
      ACNT(I,J)=0.
  150 CONTINUE
  151 CONTINUE
      DO 153 I=1,NJD
      NMEM(I)=0
  153 IBAD(')=0
C
C     READ CASE DATA AND MOVE.
 154  READ(IMT) NW1,(KE(I+1),I=1,NW1),NW2,(KD(I),I=1,NW2),NW3,
     1 (KB(I),I=1,NW3)
      IF(KE(2).EQ.KEND) GO TO 210
      KE(1)=NW1
      KE(NW1+2)=NW2
      DO 155 I=1,NW2
      J=NW1+2+I
  155 KE(J)=KD(I)
C
C     TEST VARIABLES FOR THIS CASE
      DO 205 IJD=1,NJD
      IF(IBAD(IJD).NE.0) GO TO 205
      NN=NNN(IJD)
      DO 160 I=1,36
  160 LBUF(I)=IVAR(I,IJD)
      N=0
C
C     LOOP FOR VARIABLES
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
          DO 185 L=1,NN
          KL=ISGT(L,IJD)
C
C         NEG. KL, INPUT VARIABLE       POS. KL, COMPUTED VARIABLE
          IF(KL.GT.0) GO TO 180
          KL=-KL
C
C         TEST VARIABLE LENGTH
          KM=IREL(KL)/INUM(3)
          IF(KM.EQ.IVL(L,IJD)) GO TO 170
          WRITE(6,27)IDSP(IJD)
   27     FORMAT(31HISPECIFIED VARIABLE LENGTH FOR ,A6,27H IN ERROR.  REPOR
     1T SKIPPED.)
          IBAD(IJD)=1
          WRITE(6,432) IVL(L,IJD), KL, KM
  432     FORMAT(2X,I6,' GIVEN FOR V',J3,'. SHOULD HAVE BEEN ',I6)
          GO TO 205
C
C         TEST IF INPUT VARIABLE IS IN RANGE
  170     KN=MOD(IREL(KL),INUM(3))+KM+1+KE(1)
          KS=KN-KM+1
          KLO=N+1
          KHI=KLO+KM
          KTST(L)=LASCMP(KE(KS),LBUF(KLO),LBUF(KHI),KM)
          N=N+KM+KM
          GO TO 185
C
C         TEST IF COMPUTED VARIABLE IS IN RANGE
  180     KTST(L)=0
          IF(LBUF(N+1).LE.KE(KL+1).AND.LBUF(N+2).GE.KE(KL+1))KTST(L)=1
          N=N+2
  185     CONTINUE
C
C         PERFORM LOGICAL COMBINATION OF VARIABLES
          JERR=0
          ITRA = 1
          CALL SELECT(IJD,KTST,$205,$187)
          ITRA = 2
          GO TO 190
  187     CONTINUE
          WRITE(6,29)IDSP(IJD)
   29     FORMAT(36HITHE AND/OR OPERATORS INCORRECT FOR ,A6,18H.  REPORT SK
     1IPPED.)
          IBAD(IJD)=1
          GO TO 205
C
C         ITRA = 2 MEANS THIS CASE SELECTED
C         SUM TIME SPENT INTO ARRAY
  190     DO 195 I=1,NW3
          ITASK=KB(I)/INUM(3)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
          PERCNT=MOD(KB(I),INUM(3))
          SUMS(ITASK,IJD)=PERCNT/1000. + SUMS(ITASK,IJD)
          ACNT(ITASK,IJD)=ACNT(ITASK,IJD)+1.
  195     CONTINUE
          NMEM(IJD) = NMEM(IJD)+1
  205     CONTINUE
          GO TO 154
C
C         READ AND SAVE TASK TITLES AND LOCATOR TABLE.
 210      READ(IMT) K,(ITSK(I),I=1,K)
          IF(ITSK(1).NE.KEND) GO TO 210
          BACKSPACE IMT
          BACKSPACE IMT
          READ(IMT) K,(ITSK(I),I=1,K)
          IBST=ITSK(K)/INUM(3)+1
          DO 220 IXX=1,IBST
          BACKSPACE IMT
  220     CONTINUE
          IREC=IBST-2
          IF(IPASS.EQ.1) CALL DISK(IBST,465)
         KTRK = 1
          DO 230 N=1,IREC
          READ(IMT) K,(ITSK(I),I=1,K)
          CALL DSKSTO (465,ITSK(1),KTRK,0)
          KTRK=KTRK+1
  230     CONTINUE
          READ(IMT) K,(IMVBUF(I),I=1,K)
          REWIND IMT
          DO 400 IJD=1,NJD
C
C         READ CONTROL,INDEX TITLE, AND TITLE CARDS AGAIN
          READ(ISCR,3) JEND,ISEQ,IHEAD,IBCD(NJD),IPRNT,IDSP(NJD),
        1 (ILAB(I),I=1,8)
          READ(ISCR,5)((HEAD(I,K),I=1,14),K=1,IHEAD)
          IF(IBAD(IJD).NE.0) GO TO 400
C
C RESTORE POINTER TABLE
         DO 108 I=1,NTASK
          INX(I)=I
 108     CONTINUE
         IF(ISEQ.EQ.1) CALL LSORTA(NTASK,TMEAN,INX)
         IF(ISEQ.EQ.2) CALL LSORT(NTASK,TMEAN,INX)
         IF(ISEQ.EQ.4) CALL LSORTA(NTASK,ACNT(1,IJD),INX)
         IF(ISEQ.EQ.5) CALL LSORT(NTASK,ACNT(1,IJD),INX)
C
C         PRINT REPORT.  WRITE BCD OUTPUT TAPE IF REQUESTED.
  272     IPASS=IBCD(IJD)
          DO 382 KK=1,IPASS
          CUM=0.
          IPAGE=1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      GO TO(500,550), IPRNT
  500 KCNT = IHEAD+13
      GO TO 600
  550 KCNT = IHEAD+16
  600 LM=IOUT(KK)
      WRITE(LM,40) (ILAB(IP),IP=1,8),IDSP(IJD),IPAGE
   40 FORMAT(1H1,8A6,29X,A6,6H PAGE ,I3,/)
      WRITE(LM,42)((HEAD(I,K),I=1,14),K=1,IHEAD)
   42 FORMAT(1X,13A6,A2)
      WRITE(LM,44) NMEM(IJD)
   44 FORMAT(34H NUMBER OF MEMBERS IN THE GROUP = ,I5,/,1H )
      GO TO (273,274), IPRNT
  273 WRITE(LM,46)
   46 FORMAT(/39X,55HAVERAGE PERCENT TIME SPENT BY MEMBERS PERFORMING..
     1......,  /39X,55HPERCENT OF MEMBERS PERFORMING......................
     2      .,  /39X,55HNUMBER OF MEMBERS PERFORMING..............       .
     3      .,  /39X,55HTASK INDEX (AVERAGE VALUE).........       .      .
     4      .,  /39X,55HTASK SEQUENCE NUMBER...........      .      .     .
     5      .,  /70X,24H.     .      .      .      .,/6H D TSK,20X,16HDUTY/TAS
     6K TITLES,28X,24H.   .      .      .      .,/1H )
      GO TO 275
  274 WRITE(LM,47)
   47 FORMAT(/39X,75HCUMULATIVE SUM OF AVERAGE PERCENT TIME SPENT BY AL
     1L MEMBERS................,/39X,75HAVERAGE PERCENT TIME SPENT BY AL
     2L MEMBERS............................      .,/39X,75HAVERAGE PERCEN
     3T TIME SPENT BY MEMBERS PERFORMING.............      .       .,/39X
     4,75HPERCENT OF MEMBERS PERFORMING.........................       .
     5      .      .,/39X,75HNUMBER OF MEMBERS PERFORMING..................
     6....      .      .      .       .,/39X,54HSTANDARD DEVIATION OF TASK
     7 INDEX..........       .       .,3(6X,1H.),/39X,'TASK INDEX (AVERAGE V
     8ALUE)',10(1H.),3(5X,1H.),3(6X,1H.),/39X,20HTASK SEQUENCE NUMBER,12
     9(1H.),3X,1H.,3(5X,1H.),3(6X,1H.),/70X,5H.    .,3(5X,1H.),3(6X,1H.))
      WRITE(LM,48)
   48 FORMAT(6H D TSK,20X,16HDUTY/TASK TITLES,28X,5H.    .,3(5X,1H.),3(6
     1X,1H.),/1H )
  275 M=0
          TOTRES = 0.0
      TOTDIF = 0.0
      TOTX = 0.0
      CUMP = 0.0
       DO 290 I=1,NTASK
       J=INX(I)
      IF(ITGRP(J,IJD).EQ.0) GO TO 290
       IF(ACNT(J,IJD).GT.0.) GO TO 1130
      PERPER = 0.0
      PERFRM = 0.0
      ALLMEM = 0.0
      GO TO 1131
 1130 CONTINUE
      TOTX = TOTX + 1.0
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      AMEM=NMEM(IJD)
      ALLMEM=SUMS(J,IJD)/AMEM
      PERFRM=SUMS(J,IJD)/ACNT(J,IJD)
      PERPER=ACNT(J,IJD)/AMEM*100.
     TOTDIF = TOTDIF + TMEAN(J)*ACNT(J,IJD)
     TOTRES = TOTRES + ACNT(J,IJD)
1131  CUM=CUM+ALLMEM
     CUMP = CUMP + PERPER
      M=M+1
      KTRK=IHVBUF(J)/INUM(3)
     IF(KTRK.GT.250) WRITE(6,879) KTRK
879  FORMAT(' INVALID TRACK DETECTED ', I6)
      ILOC=MOD(IHVBUF(J),INUM(3))
      CALL DSKRED(465,ITSK(1),KTRK,0)
      ISTART=ILOC+1
      ISTOP=ISTART+10
      IRCDS=(ITSK(ILOC)/INUM(3)-(ITSK(ILOC)-ITSK(ILOC)/INUM(3)*INUM(3)
     1))/11
      ILINES=IRCDS
      JCNT=ACNT(J,IJD)
      GO TO(276,277), IPRNT
  276 WRITE(LM,57)(ITSK(IX),IX=ISTART,ISTOP),M,TMEAN(J),JCNT,PERPER,
     1 PERFRM
   57 FORMAT(1H ,11A6,I4,F6.2,I5,2F7.2)
      GO TO 278
  277 WRITE(LM,58)(ITSK(IX),IX=ISTART,ISTOP),M,TMEAN(J),SD(J),JCNT,
     1 PERPER,PERFRM,ALLMEM,CUM
   58 FORMAT(1H ,11A6,I4,2F6.2,I5,4F7.2)
  278 IF(IRCDS.EQ.1) GO TO 280
      IRCDS=IRCDS-1
      ISTART=ISTART+11
      ISTOP=ISTOP+11
      WRITE(LM,59)(ITSK(IX),IX=ISTART,ISTOP)
   59 FORMAT(1H ,11A6)
      GO TO 278
  280 KCNT=KCNT+ILINES
      IF(KCNT.LT.55) GO TO 290
     IF(I.EQ.NTASK) GO TO 382
      KCNT=2
      IPAGE = IPAGE+1
      WRITE(LM,40) (ILAB(IP),IP=1,8),IDSP(IJD),IPAGE
  290 CONTINUE
  382 CONTINUE
C
     AVEDIF = TOTDIF/TOTRES
     AVEPER = CUMP/TOTX
     WRITE(ISC1) IJD,IDSP(IJD),ILAB,CUM,AVEPER,AVEDIF,TOTX,KOUNT(IJD),
    1 NMEM(IJD)
C     PUNCH REQUEST CARD IF REPORT WRITTEN ON BCD TAPE.
  385 IF(IBCD(IJD).NE.2) GO TO 400
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          IUSED=1
          CALL REPEND(ISID),IDSP(IJD),ILAB(1),JSEQ)
    400   CONTINUE
C
C         GET SOME MORE CONTROL CARDS IF NO END CARD ENCOUNTERED.
    405   IF(LEND.EQ.IEND) GO TO 407
          READ(IMT) K
          GO TO 107
    407   RETURN
          END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      COMMON KE(500),KD(1726),TMEAN(1726),SD(1726),MOPT,ACASE,
     1 IFMTWK(3452),IDATWK(3452),ISID1
      DIMENSION INFMT(71),INDAT(71),SUMX(1726),SUMX2(1726),ACNT(1726),
     1 ADATWK(3452),KFMT(12)
      DIMENSION AY(6600)
      EQUIVALENCE (ADATWK,IDATWK),(TMEAN,SUMX),(SD,SUMX2),
     1 (INFMT,INDAT),(ACNT,KD)
      DIMENSION INFILE(1600), MBUF(14)
      DATA (INFILE(I),I=1,11)/'11',6*' ',751,14,'IDSNRP',' '/
      DATA ITEE/6HT     /,IBLANK/6H      /
      CALL SETUP(10,11,26,0,0,0,20,0,0,0,0,JSEQ)
      NAVAIL=6600
      ICON=2**30
      INP=5
      ASSIGN 600 TO INPUT
      DO 5 I=1,3452
      IFMTWK(I)=0
      IDATWK(I)=0
    5 CONTINUE
      DO 6 I=1,1726
      SUMX(I)=0.
      SUMX2(I)=0.
      ACNT(I)=0.
    6 CONTINUE
C
C     READ I.D. CARD
      READ 1, ISID1,ISID2,MOPT,IPNCH,NCARD,MIN,MAX,NCASE
    1 FORMAT(7X,A6,A2,I2,I2,I3,X,A1,I2,I5)
      IF(MAX.EQ.0) MAX = 7
      AMAX=MAX
      IF(MIN.EQ.IBLANK) GO TO 180
      CALL BCDBIN(MIN,MIN,1,$200)
      GO TO 190
  180 MIN=1
  190 AMIN=MIN
C
C     READ COMMUNICATION REGION
      IHT=10
      REWIND IHT
      READ(IHT) K,(KE(I),I=1,K)
      IF(ISID1.EQ.KE(1).AND.ISID2.EQ.KE(2)) GO TO 4
      WRITE (6,14)
   14 FORMAT (38H1STUDY I.D. NO MATCH.  RUN TERMINATED.)
      GO TO 120
    4 NTASK=KE(12)
      ACASE=KE(3)
      IMAX=NAVAIL/(NTASK*2+75)
    9 IF(MOPT.LT.1.OR.MOPT.GT.3) GO TO 89
      GO TO (11,15,12),MOPT
   11 INP=11
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          ASSIGN 610 TO INPUT
          GO TO 15
    12    READ 87,(KFMT(I),I=1,12)
    87    FORMAT(12A6)
          READ(0,88) NVALS
    88    FORMAT(10X,I2)
          LNUM=0
          IPT=1
    90    KPT=IPT+NVALS-1
          IF(KPT.GT.NTASK) KPT=NTASK
          READ KFMT,KNUM,X,X,(TMEAN(I),I=IPT,KPT)
          IF(KNUM.LE.LNUM) GO TO 91
          LNUM=KNUM
          IPT=KPT+1
          IF(KPT.NE.NTASK) GO TO 90
          GO TO 112
    91    PRINT 95
    95    FORMAT('1WEIGHT CARDS OUT OF SEQUENCE.  RUN ABORTED')
          STOP
   200    ICOL = 24
          GO TO 201
    89    ICOL = 17
   201    WRITE(6,32) ICOL
    32    FORMAT('1COLUMN ',I2,' OF I.D. CARD IN ERROR.  RUN ABORTED')
          GO TO 120
    15    NCOL=NCARD*71
          J=1
C
C         READ AND STACK FORMAT CARDS
          DO 10 I=1,NCARD
          READ 2, (INFMT(N),N=1,71)
     2    FORMAT(A4,69A1,I2)
          DO 20 K=1,71
          IFMTWK(J)=INFMT(K)
          J=J+1
    20    CONTINUE
    10    CONTINUE
C
C         READ AND STACK RATER DATA
    16    DO 100 IX=1,NCASE
          J=1
          GO TO INPUT
   600    CONTINUE
          DO 30 I=1,NCARD
          READ(INP,3)(INDAT(N),N=1,71)
     3    FORMAT(A4,69A1,I2)
          DO 40 K=1,71
          IDATWK(J)=INDAT(K)
          J=J+1
    40    CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
    30  CONTINUE
        GO TO 650
  610   DO 630 I = 1, NCARD
        CALL READF(INFILE,MBUF,$650)
        DECODE(3,MBUF) (INDAT(N),N=1,71)
        DO 640 K = 1, 71
        IDATWK(J) = INDAT(K)
        J = J + 1
  640   CONTINUE
  630   CONTINUE
  650   CONTINUE
        K=1
        ICASE=IDATWK(1)
        DO 50 I=1,NCARD
C
C       CHECK CASE CONTROL AND CARD NUMBER
        IF(IDATWK(K).EQ.ICASE.AND.IDATWK(K+70).EQ.IFMTWK(K+70)) GO TO 45
        GO TO 51
   45   K=K+71
   50   CONTINUE
        K=0
        DO 60 I=1,NCOL
C
C       IF FORMAT COLUMN IS T CHECK DATA COLUMN FOR VALID RESPONSE
        IF(IFMTWK(I).NE.ITEE) GO TO 60
        K=K+1
        CALL BCDBIN(IDATWK(I),IDATWK(I),1,$56)
        ADATWK(I)=IDATWK(I)
        IF(ADATWK(I).LT.AMIN.OR.ADATWK(I).GT.AMAX) GO TO 55
        SUMX(K)=SUMX(K)+ADATWK(I)
        SUMX2(K)=SUMX2(K)+ADATWK(I)**2
        ACNT(K)=ACNT(K)+1.
        GO TO 60
   55   IF(ADATWK(I).EQ.0.) GO TO 60
   56   WRITE(6,7) IDATWK(I),K
    7   FORMAT(' CASE ',A4,' HAS RATING ',I3,' OUT OF RANGE.  RATING SKIP
       1PED.')
   60   CONTINUE
        IF(K.EQ.NTASK)GO TO 100
          IDIF = K - NTASK
        WRITE(6,46) K,NTASK,IDIF
   46   FORMAT(' THE NUMBER OF T''S IN FORMAT DOES NOT MATCH NUMBER F',
       1'ROM COMM REGION.',/,1X,I6,' T''S PRESENT',/,1X,I6,' = NTASK',/,
       1 1X,I6,' = DIFFERENCE',/,' RUN TERMINATED.')
        GO TO 120
  100   CONTINUE
C
C       COMPUTE MEANS, SD
        DO 110 I=1,NTASK
        SD(I)=SQRT((SUMX2(I)*ACNT(I)-SUMX(I)**2))/ACNT(I)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          TMEAN(I)=SUMX(I)/ACNT(I)
   110    CONTINUE
C
C         PUNCH MEANS IF REQUESTED
          IF(IPNCH.EQ.0) GO TO 112
          WRITE(1,42)
    42    FORMAT('(I2,A2,A4,12F6.2)')
          ICARD=1
          IPT=1
   111    KPT=IPT+11
          IF(KPT.GT.NTASK)KPT=NTASK
          WRITE(1,44) ICARD,ISID1,(TMEAN(I),I=IPT,KPT)
    44    FORMAT(I2,'TX',A4,12F6.2)
          ICARD=ICARD+1
          IPT=KPT+1
          IF(KPT.NE.NTASK) GO TO 111
          GO TO 112
    51    WRITE(6,52)IDATWK(K)
    52    FORMAT(15H1ERROR IN CASE ,A4,28H OF MATER DECK.  RUN ABORTED)
          GO TO 120
C
C         SET UP STORAGE FOR VARIABLE DIMENSIONS
   112    N1 = 9*IMAX+1
          N2 = N1+36*IMAX
          N3 = N2+8*IMAX
          N4 = N3+9*IMAX
          N5 = N4+IMAX
          N6 = N5+NTASK*IMAX
          N7 = N6+NTASK*IMAX
          N8 = N7+IMAX
          N9 = N8+IMAX
          N10 = N9+IMAX
          N11 = N10+IMAX
          LGM=N11+IMAX
          CALL TSKNL1 (IMAX,NTASK,AY,AY(N1),AY(N2),AY(N3),AY(N4),AY(N5),
         1 AY(N6),AY(N7),AY(N8),AY(N9),AY(N10), LGM,JSEQ)
          WRITE(6,130)
   120    STOP
   130    FORMAT('1')
          SUBROUTINE TSKNL1 (IMAX,NTASK,IVL,IVAR,IOPN,ISGT,NNN,ACNT,SUMS,
         1 IDSP,IBAD,IBCD,NMEM,LGM,JSEQ)
          COMMON KE(500),KD(1726),TMEAN(1726),SD(1726),MOPT,ACASE,
         1 IFMTWK(3452),IDATWK(3452),ISID1
          DIMENSION IVL(9,IMAX),IVAR(36,IMAX),IOPN(8,IMAX),ISGT(9,IMAX),
         1 NNN(IMAX),ACNT(NTASK,IMAX),IBCD(IMAX),IDSP(IMAX),IBAD(IMAX),
         2 SUMS(NTASK,IMAX),NMEM(IMAX)
          DIMENSION IREL(1726),HEAD(14,6),IBUF(2,2),KTST(9),
         1 INUM(5),  KB(1726),ILAB(8),FBUF(2,2),ITSK(1726),IMVBUF(1726),
         2 INX(1726),IOUT(2),LBUF(54),MMIN(1726),MEAN(1726)
          EQUIVALENCE (IBUF,FBUF),       (INX(1),IFMTWK(1)),
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      1 (IHVBUF(1),KB(1),IFMTWK(1727)),         (IREL(1),I[illegible]1),
      2 IDATWK(1)),(MMIN(1),IDATWK(1727)),(MEAN(1),KD(1))
        DATA IEND/6HEND   /,KEND/6HEND OF/
       IPASS = 0
        REAL MEAN
        IHT=10
        ISCR=20
        INUM(1)=2**30
        INUM(2)=2**24
        INUM(3)=2**18
        INUM(4)=2**12
        INUM(5)=2**6
        IOUT(1)=6
        IOUT(2)=26
        KTASK=NTASK-1
        NDICT=KE(4)
C
C       READ RELATIVE LOCATION TABLE.
  107   READ(IHT) K,(IREL(I),I=1,K)
        IUSED=0
        REWIND ISCR
C
C       SKIP DICTIONARY.
        DO 110 I=1,NDICT
  110   READ(IHT) K
C
C       READ AND CHECK CONTROL CARD.
        NJD=1
  112   READ(5,3,END=113) JEND,ISEQ,IHEAD,IBCD(NJD),IPRNT,IDSP(NJD),
      1 (ILAB(I),I=1,8)
    3   FORMAT(A3,I1,3I2,1X,A6,1X,8A6)
        IF(MOPT.EQ.3) IPRNT=1
        IF(IBCD(NJD).EQ.0) IBCD(NJD)=2
        IF(ISEQ.GE.1.AND.ISEQ.LE.3) GO TO 115
        WRITE(6,15)IDSP(NJD)
   15   FORMAT(29H1INCORRECT SORT SEQUENCE FOR ,A6,13H.  ASSUMED 1.)
        ISEQ=1
        GO TO 115
  113   LEND=IEND
        IF(NJD.GT.1) GO TO 141
        GO TO 407
  115   IF(IHEAD.GE.1.AND.IHEAD.LE.6) GO TO 120
        WRITE(6,17)IDSP(NJD)
   17   FORMAT(41H1THE NUMBER OF TITLE CARDS SPECIFIED FOR ,A6,27H IN ERR
     1OR.  RUN TERMINATED.)
        GO TO 407
  120   IF(IPRNT.EQ.1.OR.IPRNT.EQ.2) GO TO 121
        WRITE(6,19) IDSP(NJD)
   19   FORMAT(40H1PRINT OPTION INCORRECTLY SPECIFIED FOR ,A6,20H.  ASSUM
     1ED OPTION 2.)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
          IPRNT = 2
    121   IF(IBCD(NJD).GE.1.AND.IBCD(NJD).LE.2) GO TO 125
          WRITE(6,21)IDSP(NJD)
     21   FORMAT(43HIBCD TAPE OPTION INCORRECTLY SPECIFIED FOR ,A6,13H.  AS
         1SUMED 1.)
          IBCD(NJD)=1
C
C         READ TITLE CARDS.
    125 READ(5,5)((HEAD(I,K),I=1,14),K=1,IHEAD)
      5   FORMAT(13A6,A2)
C
C         READ VARIABLE NAMES AND OPERATORS
          JERR=0
         IPRNTS = 0
         CALL SAMPLE(NJD,ISGT(1,NJD),NN,JERR,IPRNTS)
         NNN(NJD) = NN
          N=1
          DO 146 K=1,NN
          IF(ISGT(K,NJD).GE.4) GO TO 135
          IF(ISGT(K,NJD).GE.0) GO TO 130
C
C         READ LIMITS - ALPHANUMERIC
          READ 9,L,KTST(K),((IBUF(J,M),M=1,L),J=1,2)
      9   FORMAT(I2,2X,I3,1X,4A6)
          KTST(K)=-KTST(K)
          GO TO 140
C
C         READ LIMITS - INTEGER
    130   READ 11,L,KTST(K),IBUF(1,1),IBUF(2,1)
     11   FORMAT(I2,2X,I3,1X,I6,6X,I6)
          GO TO 140
C
C         READ LIMITS - FLOATING POINT
    135   READ 13,KTST(K),FBUF(1,1),FBUF(2,1)
     13   FORMAT(4X,I3,1X,2F12.6)
          L = 1
  140    KL = L*2
          IF(ISGT(K,NJD).NE.KTST(K))JERR=JERR+1
C
C         TEST TO SEE IF ANY ERRORS IN VARIABLE ACTION CARDS
          IF(JERR.EQ.0) GO TO 142
          WRITE(6,23)IDSP(NJD)
     23   FORMAT(54H1ERROR IN VARIABLE INTERACTION OR DEFINITION CARD FOR ,
         1A6,15H REPORT SKIPPED)
          GO TO 112
    142   DO 145 M=1,L
          J=N+M-1
          IVAR(J,NJD)=IBUF(1,M)
          J=J+L
    145   IVAR(J,NJD)=IBUF(2,M)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
            N=N+KL
            IVL(K,NJD)=L
   146  CONTINUE
C
C       WRITE CONTROL,INDEX TITLE, AND TITLE CARDS
        WRITE(ISCR,3) JEND,ISEQ,IHEAD,IBCD(NJD),IPRNT,IUSP(NJD),
     1  (ILAB(I),I=1,8)
        WRITE(ISCR,5)((HEAD(I,K),I=1,14),K=1,IHEAD)
C
C       TEST IF MAX NUMBER OF GROUPS HAS BEEN REACHED.
        IF(NJD.GE.IMAX) GO TO 147
        NJD=NJD+1
        GO TO 112
C
   141  NJD=NJD-1
   147  CONTINUE
       IPASS = IPASS + 1
C
C       CLEAR TABLES
  157  REWIND ISCR
        DO 151 J=1,NJD
        DO 150 I=1,NTASK
        SUMS(I,J)=0.
        ACNT(I,J)=0.
   150  CONTINUE
   151  CONTINUE
        DO 153 I=1,NJD
        NMEM(I)=0
   153  IBAD(I)=0
C
C       READ CASE DATA AND MOVE.
   154  READ(IMT) NW1,(KE(I+1),I=1,NW1),NW2,(KD(I),I=1,NW2),NW3,
     1  (KB(I),I=1,NW3)
        IF(KE(2).EQ.KEND) GO TO 210
        KE(1)=NW1
        KE(NW1+2)=NW2
        DO 155 I=1,NW2
        J=NW1+2+I
   155  KE(J)=KD(I)
C
C       TEST VARIABLES FOR THIS CASE
        DO 205 IJD=1,NJD
        IF(IBAD(IJD).NE.0) GO TO 205
        NN=NNN(IJD)
        DO 160 I=1,36
   160  LBUF(I)=IVAR(I,IJD)
        N=0
C
C       LOOP FOR VARIABLES
        DO 185 L=1,NN
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          KL=ISGT(L,IJD)
C
C         NEG. KL, INPUT VARIABLE        POS. KL, COMPUTED VARIABLE
          IF(KL.GT.0) GO TO 180
          KL=-KL
C
C         TEST VARIABLE LENGTH
          KM=IREL(KL)/INUM(3)
          IF(KM.EQ.IVL(L,IJD)) GO TO 170
          WRITE(6,27)IDSP(IJD)
     27   FORMAT(31HISPECIFIED VARIABLE LENGTH FOR ,A6,27H IN ERROR.  REPOR
         1T SKIPPED.)
          IBAD(IJD)=1
          WRITE(6,432) IVL(L,IJD), KL, KM
    432   FORMAT(2X,I6,' GIVEN FOR V',J3,', SHOULD HAVE BEEN ',I6)
          GO TO 205
C
C         TEST IF INPUT VARIABLE IS IN RANGE
    170   KN=MOD(IREL(KL),INUM(3))+KM+1+KE(1)
          KS=KN-KM+1
          KLO=N+1
          KHI=KLO+KM
          KTST(L)=LASCMP(KE(KS),LBUF(KLO),LBUF(KHI),KM)
          N=N+KM+KM
          GO TO 185
C
C         TEST IF COMPUTED VARIABLE IS IN RANGE
    180   KTST(L)=0
          IF(LBUF(N+1).LE.KE(KL+1).AND.LBUF(N+2).GE.KE(KL+1))KTST(L,=1
          N=N+2
    185   CONTINUE
C
C         PERFORM LOGICAL COMBINATION OF VARIABLES
          JERR=0
          ITRA = 1
          CALL SELECT(IJD,KTST,$205,$187)
          ITRA = 2
          GO TO 190
    187   CONTINUE
          WRITE(6,29)IDSP(IJD)
     29   FORMAT(36HITHE AND/OR OPERATORS INCORRECT FOR ,A6,18H.  REPORT SK
         1IPPED.)
          IBAD(IJD)=1
          GO TO 205
C
C         ITRA = 2 MEANS THIS CASE SELECTED
C         SUM TIME SPENT INTO ARRAY
    190   DO 195 I=1,NW3
          ITASK=KB(I)/INUM(3)
          PERCNT=MOD(KB(I),INUM(3))
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          SUMS(ITASK,IJD)=PERCNT/1000. + SUMS(ITASK,IJD)
          ACNT(ITASK,IJD)=ACNT(ITASK,IJD)+1.
  195     CONTINUE
          NMEM(IJD) = NMEM(IJD)+1
  205     CONTINUE
          GO TO 154
C
C         READ AND SAVE TASK TITLES AND LOCATOR TABLE.
  210     READ(IMT) K,(ITSK(I),I=1,K)
          IF(ITSK(1).NE.KEND) GO TO 210
          BACKSPACE IMT
          BACKSPACE IMT
          READ(IMT) K,(ITSK(I),I=1,K)
          IBST=ITSK(K)/INUM(3)+1
          DO 220 IXX=1,IBST
          BACKSPACE IMT
  220     CONTINUE
          IREC=IBST-2
          IF(IPASS.EQ.1) CALL DISK(IBST,465)
         KTRK = 1
          DO 230 N=1,IREC
          READ(IMT) K,(ITSK(I),I=1,K)
          CALL DSKSTO (465,ITSK(1),KTRK,0)
          KTRK=KTRK+1
  230     CONTINUE
          READ(IMT) K,(IHVBUF(I),I=1,K)
          REWIND IMT
          DO 400 IJD=1,NJD
C
C         READ CONTROL,INDEX TITLE, AND TITLE CARDS AGAIN
          READ(ISCR,3) JEND,ISEQ,IHEAD,IBCD(NJD),IPRNT,IDSP(NJD),
     1    (ILAB(I),I=1,8)
          READ(ISCR,5)((HEAD(I,K),I=1,14),K=1,IHEAD)
          IF(IBAD(IJD).NE.0) GO TO 400
C
C         RESTORE MEAN TABLE
          DO 108 I=1,NTASK
          INX(I)=I
  108     MEAN(I)=TMEAN(I)
          GO TO(250,250,272),ISEQ
C
C         SORT ASCENDING/DESCENDING BY MEAN.
  250     IMORE=0
          DO 270 I=1,KTASK
          GO TO(260,265),ISEQ
  260     IF(MEAN(I).LE.MEAN(I+1)) GO TO 270
          GO TO 266
  265     IF(MEAN(I).GE.MEAN(I+1)) GO TO 270
  266     TEMP=MEAN(I)
          MEAN(I)=MEAN(I+1)
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          MEAN(I+1)=TEMP
          TEMP=INX(I)
          INX(I)=INX(I+1)
          INX(I+1)=TEMP
          IMORE=1
    270   CONTINUE
          IF(IMORE.EQ.0) GO TO 272
          GO TO 250
C
C         PRINT REPORT.  WRITE BCD OUTPUT TAPE IF REQUESTED.
    272   IPASS=IBCD(IJD)
          DO 382 KK=1,IPASS
          CUM=0.
          IPAGE=1
          GO TO(500,550), IPRNT
    500   KCNT = IHEAD+13
          GO TO 600
    550   KCNT = IHEAD+16
    600   LM=IOUT(KK)
          WRITE(LM,40) (ILAB(IP),IP=1,8),IDSP(IJD),IPAGE
     40   FORMAT(1H1,8A6,29X,A6,6H PAGE ,I3,/)
          WRITE(LM,42)((HEAD(I,K),I=1,14),K=1,IHEAD)
     42   FORMAT(1X,13A6,A2)
          WRITE(LM,44) NMEM(IJD)
     44   FORMAT(34H NUMBER OF MEMBERS IN THE GROUP = ,I5,/,1H )
          GO TO (273,274), IPRNT
    273   WRITE(LM,46)
     46   FORMAT(/39X,55HAVERAGE PERCENT TIME SPENT BY MEMBERS PERFORMING..
         1......, /39X,55HPERCENT OF MEMBERS PERFORMING...................
         2     ., /39X,55HNUMBER OF MEMBERS PERFORMING..............     .
         3     ., /39X,55HTASK INDEX (AVERAGE VALUE)..........   .     .
         4     ., /39X,55HTASK SEQUENCE NUMBER............   .     .     .
         5     ., /70X,24H.    .    .    .     ../6H D TSK,20X,16HDUTY/TAS
         6K TITLES,28X,24H.    .    .    .     ../1H )
          GO TO 275
    274   WRITE(LM,47)
     47   FORMAT(/39X,75HCUMULATIVE SUM OF AVERAGE PERCENT TIME SPENT BY AL
         1L MEMBERS................,/39X,75HAVERAGE PERCENT TIME SPENT BY AL
         2L MEMBERS..........................     .,/39X,75HAVERAGE PERCEN
         3T TIME SPENT BY MEMBERS PERFORMING.............    .     .,/39X
         4,75HPERCENT OF MEMBERS PERFORMING.........................     .
         5    .      .,/39X,75HNUMBER OF MEMBERS PERFORMING.................
         6....    .     .     .     .,/39X,54HSTANDARD DEVIATION OF TASK
         7 INDEX...........    .     .,3(6X,1H.),/39X,'TASK INDEX (AVERAGE V
         8ALUE)',10(1H.),3(5X,1H.),3(6X,1H.),/39X,20HTASK SEQUENCE NUMBER,12
         9(1H.),3X,1H.,3(5X,1H.),3(6X,1H.),/70X,5H.   .,3(5X,1H.),3(6X,1H.))
          WRITE(LM,48)
     48   FORMAT(6H D TSK,20X,16HDUTY/TASK TITLES,28X,5H.   .,3(5X,1H.),3(6
         1X,1H.),/1H )
    275   M=0
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          DO 290 I=1,NTASK
          J=INX(I)
          IF(ACNT(J,IJD).LE.0.) GO TO 290
          AMEM=NMEM(IJD)
          ALLMEM=SUMS(J,IJD)/AMEM
          PERFRM=SUMS(J,IJD)/ACNT(J,IJD)
          PERPER=ACNT(J,IJD)/AMEM*100.
          CUM=CUM+ALLMEM
          M=M+1
          KTRK=IHVBUF(J)/INUM(3)
         IF(KTRK.GT.250) WRITE(6,879) KTRK
  879   FORMAT(' INVALID TRACK DETECTED ', I6)
          ILOC=MOD(IHVBUF(J),INUM(3))
          CALL DSKRED(465,ITSK(1),KTRK,0)
          ISTART=ILOC+1
          ISTOP=ISTART+10
          IRCDS=(ITSK(ILOC)/INUM(3)-(ITSK(ILOC)-ITSK(ILOC)/INUM(3)*INUM(3)
       1))/11
          ILINES=IRCDS
          JCNT=ACNT(J,IJD)
          GO TO(276,277), IPRNT
  276   WRITE(LM,57)(ITSK(IX),IX=ISTART,ISTOP),M,MEAN(I),JCNT,PERPER,
       1 PERFRM
   57   FORMAT(1H ,11A6,I4,F6.2,I5,2F7.2)
          GO TO 278
  277   WRITE(LM,58)(ITSK(IX),IX=ISTART,ISTOP),M,MEAN(I),SD(J),JCNT,
       1 PERPER,PERFRM,ALLMEM,CUM
   58   FORMAT(1H ,11A6,I4,2F6.2,I5,4F7.2)
  278   IF(IRCDS.EQ.1) GO TO 280
          IRCDS=IRCDS-1
          ISTART=ISTART+11
          ISTOP=ISTOP+11
          WRITE(LM,59)(ITSK(IX),IX=ISTART,ISTOP)
   59   FORMAT(1H ,11A6)
          GO TO 278
  280   KCNT=KCNT+ILINES
          IF(KCNT.LT.55) GO TO 290
         IF(I.EQ.NTASK) GO TO 382
          KCNT=2
          IPAGE = IPAGE+1
          WRITE(LM,400) (ILAB(IP),IP=1,8),IDSP(IJD),IPAGE
  290   CONTINUE
  382 CONTINUE
C
C         PUNCH REQUEST CARD IF REPORT WRITTEN ON BCD TAPE.
  385   IF(IBCD(IJD).NE.2) GO TO 400
          IUSED=1
          CALL REPEND(ISID1,IDSP(IJD),ILAB(1),JSEQ)
  400   CONTINUE
C
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C        GET SOME MORE CONTROL CARDS IF NO END CARD ENCOUNTERED.
   405   IF(LEND.EQ.IEND) GO TO 407
         READ(IMT) K
         GO TO 107
   407   RETURN
         END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
      DIMENSION KBUF(15000),ILOC(14),JLOC(14),WINPUT(1700)
      DIMENSION JA(1),W(1),          ID(2),ITEST(2),IFMT(108)
      DIMENSION NWSA(100),SFAC(100)
      EQUIVALENCE (KBUF,W),(W,JA)
      COMMON NHIST
       DATA NAVAIL/15000/,ITEST/12HEND OF DATA /
      DATA ILOC/84HVARIABLE DICTIONARY INPUT TO VARGEN
     1                                   /
      DATA JLOC/84HVARIABLE DICTIONARY CREATED BY VARGEN
     1                                   /
      DATA IEND/6HEND   /
C
C***** KBUF    - STORAGE AREA
C***** JA      - COMMUNICATION REGION
C***** W       - WEIGHTS
C***** IFMT    - FORMAT FOR READING WEIGHTS
C***** NT1     - OLD HISTORY DATA TAPE UNIT
C***** NT2     - NEW HISTORY DATA TAPE UNIT
C***** NCV1    - OLD NO. OF C.V.
C***** NCV2    - NO. OF C.V. TO BE FORMED
C***** NCV3    -TOTAL NO. OF C.V.
C***** NCASE   - NO. OF CASES
C***** NTASK   - NO. OF TASKS
C***** NSW     -NO. OF SETS OF WEIGHTS
C***** NFC     - NO. OF FORMAT CARDS
C***** NHW     - NO. OF WORDS OF HISTORY DATA
C
       CALL SETUP(10,15,0,0,0,0,0,0,0,0,0,USEQ)
       NT1=10
       NT2=15
      READ 912, NMST1,NMST2,NCV2
  912 FORMAT(7X,A6,A2,I3)
C     PRINT THE VARIABLE DICTIONARY
      REWIND NT1
      REWIND NT2
      READ(NT1) X,X,X,X,NDICT
      READ(NT1)
      CALL PRTDIC(NT1,6,1,ILOC,NDICT)
      REWIND NT1
C
C***** COPY COMMUNICATION REGION
C
      READ(NT1)N,(JA(I),I=1,N)
      NDUTY=JA(9)
      IF(NMST1.EQ.JA(1).AND.NMST2.EQ.JA(2)) GO TO 920
      PRINT 911
  911 FORMAT('1STUDY ID NO MATCH.  RUN ABORTED.')
      CALL EXTRAN(2)
  920 CONTINUE
      NCASE=JA(3)
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      NTASK=JA(12)
      NCV1=JA(4)
      JA(4)=JA(4)+(NCV2+15)/16
      WRITE(NT2)N,(JA(I),I=1,N)
C
C***** CALL ROUTINE TO COPY DICTIONARY
C
      CALL DICTRW(JA,KBUF(51),KBUF(1051),KBUF(1077),KBUF(1317),NT1,NT2,
     1 NCV1,NCV2)
      NSW=0
      READ(NT1)K,(DUMMY,I=1,K),NHW
      BACKSPACE NT1
      NCV3=NCV1+NCV2
      NTEST=NCV3+NHW
      IF(NTEST.LE.1500)GO TO 7
      WRITE(6,906)NTEST
  906 FORMAT( 59H NO. OF COMPUTED VARIABLES PLUS NO. OF HISTORY VARIABLE
     1S = I4,13H RUN ABORTED.  )
      CALL ERTRAN(2)
    7 CONTINUE
C
C***** READ SETS OF WEIGHTS
C
C***** SET UP VARIABLE STORAGE
C
C
      IX=1
 1    READ 901, ID(1),STP
 901  FORMAT(A6,X,F10.0)
      IF(ID(1).EQ.IEND) GO TO 4
      NSW=NSW+1
      NWSA(NSW)=ID(1)
       IF (STP.EQ.0.0) STP = 1.0
       SFAC(NSW) =STP
      READ 903, (IFMT(I),I=1,12)
  903 FORMAT(12A6)
       READ(5,800) NVALS
 800  FORMAT(10X,I2)
      NX1=1+NTASK*NSW
      NX2=NX1+NCV3
      NX3=NX2+NHW
      NX4=NX3+NTASK
      NX5=NX4+NCV2
      NX6=NX5+NCV2
      NX7=NX6+NCV2
      IF(NX7+6*NSW.LE.NAVAIL) GO TO 809
      WRITE(6,904)
  904 FORMAT(72H1 DATA REQUIRES TOO MUCH STORAGE.  PROCESS THE DATA IN T
     1WO OR MORE RUNS.)
       CALL ERTRAN(2)
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 809  LNUM=0
      IPT=1
 810  KPT = IPT+NVALS-1
      IF(KPT.GT.NTASK) KPT=NTASK
      READ IFMT,KNUM,X,X,(WINPUT(I),I=IPT,KPT)
      IF(KNUM.LE.LNUM) GO TO 820
      LNUM = KNUM
      IPT=KPT+1
      IF(KPT.NE.NTASK) GO TO 810
      GO TO 830
 820  PRINT 825
 825  FORMAT('1WEIGHT DECK OUT OF SEQUENCE.  RUN ABORTED.')
      CALL ERTRAN(2)
 830  DO 840 I=1,NTASK
      W(IX)=WINPUT(I)
 840  IX=IX+1
      GO TO 1
C
C***** APPLY WEIGHTS TO CASE DATA
C
    4 CALL APLWTS(W,KBUF(NX1),KBUF(NX2),KBUF(NX3),KBUF(NX4),KBUF(NX5),
     1 KBUF(NX6),KBUF(NX7),NT1,NT2,NCV1,NCV2,NCV3,NS0,NCASE,NTASK,NHW,
     2 NWSA,SFAC)
C
C***** COPY REMAINDER OF HISTORY DATA TAPE
C
      READ(NT1) N1, (KBUF(I),I=1,N1)
      IF(KBUF(1).EQ.ITEST(1).AND.KBUF(2).EQ.ITEST(2))GO TO 5
      WRITE(6,905)
  905 FORMAT(99H1 NCASE DIFFERS FROM ACTUAL NUMBER OF CASES ON THE HISTO
     1RY DATA TAPE.  CHECK THE HISTORY DATA TAPE.)
      CALL ERTRAN(2)
    5 N2 = 1
      N3 = 0
      WRITE(NT2) N1, (KBUF(I),I=1,N1), N2,N3,N2,N3
    6 READ(NT1)N1,(KBUF(I),I=1,N1)
      WRITE(NT2)N1,(KBUF(I),I=1,N1)
      IF(KBUF(1).NE.ITEST(1).OR.KBUF(2).NE.ITEST(2))GO TO 6
      REWIND NT1
      ENDFILE NT2
      REWIND NT2
      READ(NT2) X,X,X,X,NDICT
      READ(NT2)
      CALL PRTDIC(NT2,6,1,JLOC,NDICT)
      REWIND NT2
      WRITE(6,907)
  907 FORMAT('1')
      STOP
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE APLWTS(W, CV,IHV,ITV,ICN,IOP,IWS,S,NT1,NT2,NCV1,NCV2,
     1 NCV3,NSW,ICASE,NTASK,NHW,NWSA,SFAC)
      DIMENSION W(NTASK,NSW), CV(NCV3),IHV(NHW),ITV(NTASK),ICN(NCV2),
     1 IOP(NCV2),IWS(NCV2),S(6,NSW)
      DIMENSION NWSA(100),SFAC(100)
      DOUBLE PRECISION ISM,ISQ
      DIMENSION ISM(100), ISQ(100)
      DATA MBIT/01000000/
C
C***** W       - WEIGHTS
C***** ICN     - COMPUTED VAR. NUMBERS
C***** IOP     - OPERATORS
C***** IWS     - WEIGHT SET NUMBERS
C***** CV      - COMPUTED VAR.
C***** IHV     - HISTORY VAR.
C***** ITV     - TASK VAR.
C***** S       - SUMMATIONS
C***** NX      - INDEX OF TASK VALUES
C***** P       - TASK VALUE(PERCENTAGE)
C***** WT      - WEIGHT CORRESPONDING TO NX
C***** KCN     - CURRENT COMPUTED VAR. NO.
C***** KOP     - CURRENT OPERATOR
C***** KWS     - CURRENT WEIGHT SET NO.
C***** NWSA    - NAME OF WEIGHT SET ARRAY
C***** NWS     - NAME OF WEIGHT SET
C
C***** READ ACTION CARDS
C
      DO 1 I=1,NCV2
      ISM(I) = 0.0
      ISQ(I) = 0.0
      READ(5,900) ICN(I),IOP(I),NWS
  900 FORMAT(1X,I3,1X,I1,1X,A6)
      IF(IOP(I).EQ.0.OR.IOP(I).GT.6) IOP(I) = 1
      IF(ICN(I).GT.NCV1.AND.ICN(I).LE.NCV3) GO TO 22
      WRITE(6,904)ICN(I)
  904 FORMAT(28H COMPUTED VARIABLE NUMBER = I4,31H IS OUT OF RANGE.  RUN
     1 ABORTED.)
      CALL ERTRAN(2)
   22 CONTINUE
      IF(I.EQ.1)GO TO 23
      I1=I-1
      DO 20 J=1,I1
      IF(ICN(I).NE.ICN(J)) GO TO 20
      WRITE(6,905)I,J
  905 FORMAT(20H ACTION CARDS NUMBERI4,5H AND I4,57H SPECIFY THE SAME CO
     1MPUTED VARIABLE NUMBER.  RUN ABORTED.)
      CALL ERTRAN(2)
   20 CONTINUE
   23 CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      DO 21 J=1,NSW
      IF(NWS.NE.NWSA(J))GO TO 21
      IWS(I)=J
      GO TO 1
   21 CONTINUE
      WRITE(6,906)NWS
  906 FORMAT(17H WEIGHT SET NAME A6,87H ON AN ACTION CARD DOES NOT MATCH
     1 ANY OF THE INPUT WEIGHT SET I.D. NAMES.  RUN ABORTED.)
      CALL ERTRAN(2)
    1 CONTINUE
      DO 2 KAS=1,NCASE
      DO 3 I=1,NCV3
3     CV(I) = 0.0
C
C***** READ CASE DATA
C
      READ(NTI)M,(CV(I),I=1,M),N,(IHV(I),I=1,N),NNZ,(ITV(I),I=1,NNZ)
      IF(NCV1.EQ.M)GO TO 4
      WRITE(6,901)
  901 FORMAT(  112H NO. OF COMPUTED VARIABLE TITLES DOES NOT EQUAL NUMBE
     1R OF COMPUTED VARIABLES ON HISTORY DATA TAPE.  RUN ABORTED.)
      CALL ERTRAN(2)
    4 IF(NHW.EQ.N)GO TO 5
      WRITE(6,902)KAS
  902 FORMAT(6H CASE I5, 77H DOES NOT HAVE THE SAME NO. OF WORDS OF HIST
     1ORY DATA AS CASE 1.  RUN ABORTED.)
      CALL ERTRAN(2)
    5 IF(NNZ.GT.0.AND.NNZ.LE.928)GO TO 6
      WRITE(6,903)KAS,NNZ
  903 FORMAT(  6H CASE I5,  5H HAS I4, 37H NON-ZERO TASK VALUES.  CASE D
     1ELETED.)
      GO TO 2
C
C***** COMPUTE SUMMATIONS
C
    6 DO 8 K=1,6
      DO 8 J=1,NSW
    8 S(K,J)=0.0
      DO 9 I=1,NNZ
      NX=ITV(I)/MBIT
      IP=MOD(ITV(I),MBIT)
      P=IP
      P=P/1000.
      DO 10 J=1,NSW
      S(5,J) = S(5,J) + ABS(P-WT)
      S(6,J) = S(6,J) + (P-WT)**2
      WT=W(NX,J)
      S(1,J)=S(1,J)+WT*P
      IF(P.GT.0.0) S(2,J)=S(2,J)+WT
   10 S(3,J)=S(3,J)+AMIN1(WT,P)
```

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      9 CONTINUE
        FNZ=NNZ
        DO 11 J=1,NSW
        S(1,J)=S(1,J)/SFAC(J)
     11 S(4,J)=S(2,J)/(FNZ)
C
C***** STORE COMPUTED VAR.
C
        DO 12 I=1,NCV2
        KCN=ICN(I)
        KOP=IOP(I)
        KWS=IWS(I)
        CV(KCN)=S(KOP,KWS)
        C = CV(KCN)
        ISM(I) = ISM(I) + C
        ISQ(I) = ISQ(I) + C*C
 12     CONTINUE
C
C***** WRITE CASE DATA ON NT2
C
        WRITE(NT2)NCV3,(CV(I),I=1,NCV3),NHW,(IHV(I),I=1,NHW),NNZ,(ITV(I),
       1I=1,NNZ)
      2 CONTINUE
        Z=NCASE
        DO 210 I = 1, NCV2
        IF(MOD(I,51).EQ.1) WRITE(6,200)
 200    FORMAT('1 VAR ID      MEAN VALUE     STD DEV',//)
        XBAR = ISM(I)/Z
        SD = SQRT(ABS(ISQ(I)/Z - XBAR*XBAR))
        WRITE(6,201) ICN(I), XBAR, SD
 201    FORMAT(4X,'C',J3,F15.3,F11.3)
 210    CONTINUE
        RETURN
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE DICTRW(JA,IHVLT,ITIDT,ICVT,IHVT,NT1,NT2,NCV1,NCV2)
      DIMENSION JA(1),IHVLT(1),ITIDT(1),ICVT(15,1),IHVT(15,1)
      COMMON NHV
      DATA IBLNK/6H      /
      DATA ICEE /1RC/
C
C***** IHVLT  - HIST. VAR. LOC. TABLE
C***** ITIDT  - TASK IN DUTY TABLE
C***** ICVT   - COMPUTED VAR. TITLES
C***** IHVT   - HIST. VAR. TITLES
C***** NHV    - NO. OF HIST. VAR.
C***** NDTY   - NO. OF DUTIES
C***** NHVTR  - NO. OF HIST. VAR. TITLE RECORDS
C***** NCVTR  - NO. OF COMPUTED VAR. TITLE RECORDS
C***** NUM    - NO. OF TITLES IN THE CURRENT GROUP OF 16
C
C***** COPY LOC- TABLE AND TASK IN DUTY TABLE
C
      READ(NT1)NHV,(IHVLT(I),I=1,NHV) ,NDTY,(ITIDT(I),I=1,NDTY)
      WRITE(NT2)NHV,(IHVLT(I),I=1,NHV),NDTY,(ITIDT(I),I=1,NDTY)
       NCVTR=0
       NCOMP=NCV1
      NCV1=0
C
C***** COPY OLD COMP. VAR. TITLES
C
    1 READ (NT1)N,((ICVT(I,J),I=1,15),J=1,N)
      IVAR=ICVT(1,1)/2**30
      IF(IVAR.NE.ICEE) GO TO 5
      WRITE(NT2)N,((ICVT(I,J),I=1,15),J=1,N)
       NCVTR=NCVTR+1
       NCV1=NCV1+N
       GO TO 1
    5  NUM=0
       BACKSPACE NT1
       NHVTR=NCOMP-NCVTR
C
C***** COPY NEW COMP. VAR. TITLES
C
      DO 2 K=1,NCV2
      NUM=NUM+1
      READ(5,900)(ICVT(I,NUM),I=1,14)
  900 FORMAT(13A6,A2)
      KFMT=ICVT(13,NUM)
      ICVT(13,NUM)=IBLNK
      ICVT(15,NUM)=KFMT
      IF(NUM.LT.16)GO TO 2
      WRITE(NT2)NUM,((ICVT(I,J),I=1,15),J=1,NUM)
      NUM=0
    2 CONTINUE
```

PROGRAM TYPE: SUBPROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      IF(NUM.GT.0)WRITE(NT2)NUM,((ICVT(I,J),I=1,15),J=1,NUM)
C
C***** COPY HIST. VAR. TITLES
C
      DO 3 K=1,NHVTR
      READ(NT1)N,((IHVT(I,J),I=1,15),J=1,N)
    3 WRITE(NT2)N,((IHVT(I,J),I=1,15),J=1,N)
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      DIMENSION FMT1(22),KFMT1(22),FMT2(22),KFMT2(22),SD(100),
     1 AMEAN(100),IVAR(100),TITLE(14),JA(50),IREL(1000),ICOMP(100),
     2 IDVAR(100),LHIST(100),TYPE(100),IVAL(100),IFMT(100),AA(1000),
     3          KB(1000),ARRAY(100),KC(15,20),A(11000),KA(11000),
     4          IVNUM(100),IBAD(100),SUM(100),SUMSQ(100)
      COMMON FMT1,FMT2,A
      EQUIVALENCE(FMT1,KFMT1),(FMT2,KFMT2),(A,KA), (A(1),IVAR(1)),
     1             (A(101),TITLE(1)), (A(115),JA(1)), (A(165),IREL(1)),
     2             (A(1165),ICOMP(1)), (A(1265),LHIST(1)),
     3             (A(1365),TYPE(1)), (A(1465),IVAL(1)),
     4             (A(1565),IFMT(1)),
     5             (A(1725),KB(1)), (A(2725),ARRAY(1)),
     6             (A(2825),KC(1,1)), (A(3125),IVNUM(1)),
     7             (A(3225),IBAD(1)),(A(3325),AA(1))
      DATA CORR/6HCORR  /, END/6HEND OF/
      DATA C/6HC     /,V/6HV     /
      DATA ICEE, IONE, ISIX / 1RC, 1R1, 1R6/
      ISHF6=2**6
      ISHF12=2**12
      ISHF18=2**18
       ISHF30=2**30
       IHT=10
       IOUT=15
       ISCR=20
       ISCR1=21
       CALL SETUP(IHT,0,0,0,0,0,ISCR,ISCR1,0,0,0,JSEQ)
      CALL MARGIN
C****
C**** NC IS THE NUMBER OF COMPUTED VARIABLE DEFINITIONS READ FROM THE
C**** CODAP HISTORY TAPE.  NV IS THE NUMBER OF HISTORY VARIABLE
C**** DEFINITIONS READ.  NHIST IS THE NUMBER OF HISTORY VARIABLES
C**** SELECTED FOR CORRELATION.  NCOMP IS THE NUMBER OF COMPUTED
C**** VARIABLES SELECTED.  NCASE IS THE NUMBER OF CASES ON THE CODAP
C**** TAPE MINUS ALL CASES WHICH HAD AN OUT OF RANGE VARIABLE.  AN OUT
C**** OF RANGE VARIABLE IS ONE WHICH CONTAINS ALPHA CHARACTERS OR AN
C**** EMBEDDED BLANK.
      NC=0
      NV=0
      NHIST=0
      NCOMP=0
      NCASE=0
C****
C**** READ PROGRAM ID CARD.
      READ 2,IPROG,ID1,ID2,ITIT
    2 FORMAT(A6,1X,A6,A2,1X,I1)
      ILINE=ITIT+3
      IF(ITIT.EQ.0) GO TO 100
      PRINT 3
    3 FORMAT('1')
C****
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C.... READ AND PRINT TITLE CARDS.
      DO 90 J=1,ITIT
      READ 4,(TITLE(I),I=1,14)
      PRINT 5,(TITLE(I),I=1, 14 )
   90 CONTINUE
    4 FORMAT(13A6,A2)
    5 FORMAT(' ',14A6)
C....
C.... READ A CARD.  IT WILL BE 'CORR' OR 'REGR'.
  100 READ(5,6) NVAR,XVAR,IXVAR,FGOOD
      IGOOD = FGOOD
      NJUMP = 1
      IF(IXVAR.GT.0) NJUMP=2
    6 FORMAT(I3,1X,A1,I3,1X,F1.0)
      IF(NVAR.LT.1.OR.NVAR.GT.100) GO TO 330
C....
C.... A 'CORR CARD WAS READ.  READ COMMUNICATION REGION AND CHECK
C.... STUDY ID.
       READ(IHT) K,(JA(I),I=1,K)
      LCASE=JA(3)
      NDICT=JA(4)
      IF(JA(1).NE.ID1.OR.JA(2).NE.ID2) GO TO 300
C....
C.... READ RELATIVE LOCATION TABLE AND SAVE.
       READ(IHT) K,(IREL(I),I=1,K)
C....
C.... READ VARIABLE SPEC CARDS.  VARIABLES CAN BE READ IN ANY ORDER,
C.... BUT WILL BE SORTED IN ASCENDING ORDER.  VARIABLE DEFINITIONS WILL
C.... BE PRINTED IN ASCENDING ORDER, COMPUTED VARIABLES FIRST.
C.... VARIABLES WILL BE WRITTEN ON TAPE(FOR THE CORRELATION ROUTINE) IN
C.... THE SAME ORDER.
      READ 8,(TYPE(I),IVAL(I),I=1,NVAR)
    8 FORMAT(16(A1,I3,X))
C....
C.... LOOK AT EACH VARIABLE READ IN.  STORE NUMBER OF COMPUTED VARIABLES
C.... READ IN LOCATION NCOMP.  STORE COUNT OF HISTORY VARIABLES IN NHIST
C.... PRINT ERROR MESSAGE IF HISTORY VARIABLE IS LARGER THAN ONE WORD.
C.... STORE COMPUTED VARIABLE NUMBERS IN ARRAY ICOMP.  STORE HISTORY
C.... VARIABLE NUMBERS IN ARRAY IVNUM.  STORE RELATIVE LOCATION OF
C.... HISTORY VARIABLES IN LHIST.  MAKE SURE THAT VARIABLES ARE
C.... SPECIFIED CORRECTLY AND IN RANGE.  CLEAR IBAD TABLE.
      DO 110 I=1,NVAR
      IBAD(I)=0
      IF(IVAL(I).EQ.0) GO TO 320
      IF(TYPE(I).EQ.V) GO TO 105
      IF(TYPE(I).NE.C) GO TO 350
      NCOMP=NCOMP+1
      ICOMP(NCOMP)=IVAL(I)
      GO TO 110
  105 NHIST=NHIST+1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      IF(IVAL(I).GT.K) GO TO 320
      IVNUM(NHIST)=IVAL(I)
      IK=IVAL(I)
      ITEST=IREL( IK )/ISHF18
      IF(ITEST.GT.1) GO TO 305
      LHIST(NHIST)=MOD(IREL( IK ),ISHF18)
  110 CONTINUE
      PRINT 9
    9 FORMAT('0')
      PRINT 10
   10 FORMAT('L',47HVAR NO      CODAP VARIABLE DEFINITION AND FORMAT)
C....
C.... SORT ARRAYS ICOMP AND LHIST ASCENDING SO THAT VARIABLES WILL BE
C.... EXTRACTED AND WRITTEN ON TAPE IN ASCENDING VARIABLE ORDER,
C.... COMPUTED VARIABLES FIRST.
      IF(NCOMP.LT.2) GO TO 113
      NUM=NCOMP-1
  111 IMORE=0
      DO 112 N=1,NUM
      IF(ICOMP(N).LE.ICOMP(N+1)) GO TO 112
      IHOLD=ICOMP(N)
      ICOMP(N)=ICOMP(N+1)
      ICOMP(N+1)=IHOLD
      IMORE=1
  112 CONTINUE
      IF(IMORE.NE.0) GO TO 111
  113 IF(NHIST.LT.2) GO TO 116
      NUM=NHIST-1
  114 IMORE=0
      DO 115 N=1,NUM
      IF(LHIST(N).LE.LHIST(N+1)) GO TO 115
      IHOLD=LHIST(N)
      LHIST(N)=LHIST(N+1)
      LHIST(N+1)=IHOLD
      IHOLD=IVNUM(N)
      IVNUM(N)=IVNUM(N+1)
      IVNUM(N+1)=IHOLD
      IMORE=1
  115 CONTINUE
      IF(IMORE.NE.0) GO TO 114
C....
C.... READ DICTIONARY CONTAINING VARIABLE DEFINITIONS.  PRINT THE
C.... DEFINITION FOR EACH VARIABLE SELECTED.  STORE THE SIZE(CHARS) OF
C.... EACH VARIABLE IN ARRAY IFMT.
  116 ISEQ=1
      DO 160 N=1,NDICT
      READ(IHT) K,((KC(I,J),I=1,15),J=1,K)
      ITYP=KC(1,1) / ISHF30
      IF(ITYP.NE.ICEE) GO TO 135
      IF(NCOMP.EQ.0) GO TO 160
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      DO 130 IX=1,K
      DO 120 KK=1,NCOMP
      IK=NC+IX
      IF(ICOMP(KK).NE.IK) GO TO 120
      PRINT 12,ISEQ,(KC(I,IX),I=1,15)
   12 FORMAT(' ',3X,I3,5X,15A6)
      ISEQ=ISEQ+1
      ILINE=ILINE+1
      IF(ILINE.NE.55) GO TO 130
      ILINE=0
      PRINT 3
      GO TO 130
  120 CONTINUE
  130 CONTINUE
      NC=NC+K
      GO TO 160
  135 IF(NHIST.EQ.0) GO TO 160
      DO 150 IX=1,K
      DO 140 KK=1,NHIST
      IK=NV+IX
      IF(IVNUM(KK).NE.IK) GO TO 140
      PRINT 12,ISEQ,(KC(I,IX),I=1,15)
      ISEQ=ISEQ+1
C****
C**** VARIABLE FORMATS MAY BE RIGHT ADJUSTED IN WORD 15 OR THEY MAY
C**** CONTAIN A TRAILING BLANK DEPENDING ON WHEN THE TAPE WAS CREATED.
      DECODE(405,KC(15,IX)) IFMT(KK)
405   FORMAT(5X,A1)
      CALL BCDBIN(IFMT(KK),IFMT(KK),1,$137)
      GO TO 138
137   DECODE(406,KC(15,IX)) IFMT(KK)
406   FORMAT(4X,A1)
      CALL BCDBIN(IFMT(KK),IFMT(KK),1,$305)
138   IF(IFMT(KK).LT.1.OR.IFMT(KK).GT.6) GO TO 305
      ILINE = ILINE + 1
      IF(ILINE.NE.55) GO TO 150
      ILINE=0
      PRINT 3
      GO TO 150
  140 CONTINUE
  150 CONTINUE
      NV=NV+K
  160 CONTINUE
C****
C**** READ CASE DATA.  EXTRACT COMPUTED VARIABLES AND BACKGROUND
C**** VARIABLES AND WRITE ON UNIT 3.  SKIP ANY CASE WHICH CONTAINS
C**** OUT OF RANGE VARIABLES.  SUBROUTINE ICVNI CONVERTS HISTORY
C**** VARIABLES FROM BCD TO BINARY.  VARIABLE DATA IS WRITTEN ON UNIT
C**** 3 SO THAT NCASE CAN BE ESTABLISHED BEFORE STARTING CORRELATION.
C**** A COMPUTED VARIABLE WHICH WAS SET OUT OF RANGE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C**** BY PREGEN WILL EQUAL -999999.
 180   READ(IMT) K,(AA(I),I=1,K),L,(KB(I),I=1,L)
      IF(AA(1).EQ. END     ) GO TO 240
      GO TO (190,185), NJUMP
  185 IF(IXVAR.GT.K) GO TO 360
      IF(AA(IXVAR).EQ.FGOOD) GO TO 190
      LCASE = LCASE-1
      GO TO 180
  190 ICHK=0
      IF(NCOMP.EQ.0) GO TO 210
      DO 200 I=1,NCOMP
      J=ICOMP(I)
      IF(J.GT.K) GO TO 340
      ARRAY(I)=AA(J)
      IF(ARRAY(I).NE.(-99999999.)) GO TO 200
      IBAD(I)=IBAD(I)+1
      ICHK=1
  200 CONTINUE
  210 IF(NHIST.EQ.0) GO TO 230
      DO 220 K=1,NHIST
      J=LHIST(K)
      I=NCOMP+K
       CALL BCDBIN(KB(J),KB(J),IFMT(K),$215)
      ARRAY(I)=KB(J)
      GO TO 220
  215 IBAD(I)=IBAD(I)+1
      ICHK=1
  220 CONTINUE
  230 IF(ICHK.EQ.1) GO TO 180
      NCASE=NCASE+1
      JCASE=NCASE+1
       WRITE(ISCR) JCASE,(ARRAY(I),I=1,NVAR)
      GO TO 180
  240 KCASE=LCASE-NCASE
      PRINT 9
      PRINT 11,NCASE,KCASE
   11 FORMAT('KCASES USED =',I5,'.  CASES REJECTED ON INVALID VARIABLES
     1=',I4,'.')
      GO TO (245,242), NJUMP
  242 PRINT 15, IXVAR,IGOOD
   15 FORMAT(' CASES WERE USED WHEN COMPUTED VARIABLE',I3,' WAS EQUAL TO
     1 ',I1,'.')
  245  REWIND ISCR
       REWIND IMT
C****
C**** PRINT A MATRIX SHOWING THE NUMBER OF INVALID VARIABLES FOUND.
      PRINT 30
   30 FORMAT('1VARIABLE ERROR COUNTS',////)
      KOUNT = 0
      IF(NCOMP.EQ.0) GO TO 3460
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
        DO 3456 I = 1, NCOMP
        ENCODE(3457,IDVAR(I)) ICOMP(I)
 3457   FORMAT('C',J3)
 3456   CONTINUE
 3460   KOUNT = NCOMP
        IF(NHIST.EQ.0) GO TO 3470
        DO 3466 I = 1, NHIST
        KOUNT = KOUNT + 1
        ENCODE(3467,IDVAR(KOUNT)) IVNUM(I)
 3467   FORMAT('V',J3)
 3466   CONTINUE
 3470   CONTINUE
        PRINT 40, (IDVAR(I),IBAD(I),I=1,NVAR)
     40 FORMAT (7(1X,A4,'.',I4,5X),/)
        IF(NCASE.EQ.0) GO TO 310
C****
C**** SET UP PERSUB TYPE TAPE HEADER AND WRITE ON UNIT 2.  WRITE NCASE
C**** RECORDS AND CLOSE TAPE WITH PERSUB TYPE END RECORD.
        DO 250 IX=1,NCASE
         READ(ISCR) JCASE,(ARRAY(I),I=1,NVAR)
        DO 250 IY = 1,NVAR
        SUM(IY) = SUM(IY) + ARRAY(IY)
        SUMSQ(IY) = SUMSQ(IY) + ARRAY(IY)**2
    250 CONTINUE
        ACASE = NCASE
        DO 4700 I = 1, NVAR
        AM = SUM(I)/ACASE
        AMEAN(I) = AM
        SD(I) = SQRT(ABS(SUMSQ(I)/ACASE - AM*AM))
 4700   CONTINUE
        DO 4735 I = 1, NVAR
        I2 = I - 1
        IF(MOD(I2,50).NE.0) GO TO 4734
        WRITE(6,4733)
 4733   FORMAT('1   VAR ID',8X,'MEAN',8X,'S.D.')
 4734   WRITE(6,4732) IDVAR(I), AMEAN(I), SD(I)
 4732   FORMAT(6X,A4,2F12.4)
 4735   CONTINUE
        CALL MARGIN
        STOP
C**** ERROR MESSAGES
    300 PRINT 14, JA(1),JA(2)
     14 FORMAT('ISTUDY ID FROM HISTORY OR KPATH FILE DOES NOT MATCH CONTRO
       1L CARD.  FILE ID IS ',A6,A2)
        GO TO 400
    305 PRINT 16, IK
     16 FORMAT(18H1HISTORY VARIABLE ,I3,95H IS NOT BETWEEN 1 AND 6 CHARACT
       1ERS ACCORDING TO THE VARIABLE DICTIONARY ENTRY.  RUN TERMINATED.)
          GO TO 400
    310 PRINT 18
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
   18 FORMAT( 99H1ALL CASE DATA REJECTED.  A BACKGROUND VARIABLE CONTAIN
     1ING ONLY ALPHA DATA MAY HAVE BEEN REQUESTED.)
      GO TO 400
  320 PRINT 20, IVAL(I)
   20 FORMAT(21H1BACKGROUND VARIABLE ,I3,34H DOES NOT EXIST ON THE CODAP
     1 TAPE.)
      GO TO 400
  330 PRINT 22
   22 FORMAT( 80H1THE NUMBER OF VARIABLES SPECIFIED ON THE VARIABLE REQU
     1EST CARD IS OUT OF RANGE.)
      GO TO 400
  340 PRINT 24, J
   24 FORMAT(18H1COMPUTED VARIABLE,I3,34H DOES NOT EXIST ON THE CODAP TA
     1PE.)
      GO TO 400
  350 PRINT 26, IVAL(I)
   26 FORMAT(17H1VARIABLE NUMBER ,I3,59H FROM THE VARIABLE REQUEST CARD
     1DOES NOT BEGIN WITH V OR C.)
      GO TO 400
  360 PRINT 24, IXVAR
400   CALL ERTRAN(2)
401   WRITE(6,370)
  370 FORMAT('1')
      STOP
      END
```

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      INTEGER R, REQCRD
      DIMENSION KOUT(2), IKILL(50)
       DIMENSION  IEND(2)
      DIMENSION JCOMM(60), IREL(928), KDATA(930), KEND(2), IHEADC(84),
     1          ITITLE(8), NMASK(580), JMASK(35), RMASK(35)
      DIMENSION  KTABLE (40000)
      DIMENSION CTABLE(1)
      EQUIVALENCE (KTABLE,CTABLE)
      DIMENSION REQCRD (69,10)
      DIMENSION AEQCRD(69,10)
      EQUIVALENCE (REQCRD, AEQCRD), (JMASK, RMASK)
      DIMENSION SUMX(10), SUMX2(10), TMBRS(10), MBRS(10)
      DIMENSION FMEAN(10), SD(10), KOTHER(10), IOTHER(10)
      DIMENSION KDICT(15,16)
      EQUIVALENCE (KDATA, KDICT), (KDATA, NMASK)
      EQUIVALENCE (R,NREQ)
      DIMENSION  DATA (1)
      EQUIVALENCE (KDATA, DATA)
      EQUIVALENCE (LO,ALO), (IHI,AHI), (KBLANK,AMASK)
      EQUIVALENCE (ICALC,CALC), (IDATA,ADATA)
      EQUIVALENCE (ISIZE, ASIZE)
      DATA KBLANK/6H      /,IEND/12HEND   END SP/,KEND/12HEND OF DATA /
      DATA KWORDC/6HC     /
      DATA KC001/6HC001 =/
      DATA IPLUS/6H++++++/
      DATA NAME/'VARSUM'/
C***** GENERATE BINARY BIT MASKS
      JMASK(1) = 1
      DO 1001 I=1,34
 1001 JMASK (I+1) = 2**I
      MMBIT = 17179869184
      MBIT = 262144
      K2P12 = 2**12
C  TAPE UNIT ASSIGNMENTS
      CALL SETUP(10,27,26,0,0,0,20,0,0,0,0,LSEQ)
      IP = 10
      IT = 27
      IR = 26
      IS = 20
      KOUT(1) = 6
      KOUT(2) = IR
      KTSIZE = 0
      REWIND IP
      REWIND IT
      ITSWT = 0
C
C PROGRAM RECYCLES HERE AFTER EACH END CARD OF A COMPLETE DATA SET.
    1   CONTINUE
      END = 0.0
      REWIND IS
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      NREPT = 0
C
      IDREAD = 0
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C
C READ COMMUNICATION REGION, FIRST K ITEMS IN FIRST RECORD
      READ (IP) K, (JCOMM(I), I=1,K)
C
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C
C READ HISTORY RELATIVE TABLE, 928 WORDS MAX, DESCRIBING RELATIVE LOC
C   OF EACH VARIABLE IN ITS RECORD.  ADD= REL LOC, DEC=NBR WORDS LENGTH
      READ (IP) K, (IREL(I), I=1,K)
      NVARS = K
C
C TEST FIRST DICTIONARY TITLE FOR COMPUTED VARIABLE IDENTIFIER.
C IF TAPE HAS COMPUTED DICTIONARY TITLES, THE FIRST TITLE WILL BE
C C001.   THIS PERMITS OLD HISTORY DATA TAPES TO BE RUN ON THIS VARSUM.
      NCB = -1
      NCC = 0
      READ (IP) K, KDATA(1)
      IF (KDATA(1) .EQ. KC001) GOTO 3
      NCB = 1
      NCC = 1000
C COPY DICT TITLES TO WORK TAPE
 3    I = JCOMM(4)
      BACKSPACE IP
      DO 2 II = 1, I
      READ (IP) K, ((KDICT(J,M), J=1,15), M=1,K)
 2    WRITE(15) K, ((KDICT(J,M), J=1,15), M=1,K)
      READ(IP) NCVAR, (KDATA(1),J=1,NCVAR), NVCNT
      REWIND IP
C
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C
C  READ VARSUM CARD
 20   NPAGE = 0
      READ(5,97,END=1000) KDATA(1), ISID1, ISID2, IHEAD
 97   FORMAT (A6,1X,A6,A2,1X,I1)
C
      IF(KDATA(1).EQ.IEND(1)) GO TO 1000
      IF(KDATA(1).EQ.NAME) GO  TO 21
      WRITE(6,99) KDATA(1), ISID1, ISID2, IHEAD
 99   FORMAT('1INVALID VARSUM CONTROL CARD = ',A6,1X,A6,A2,1X,I1)
C* • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C
C  THIS SECTION BYPASSES CARDS AFTER ERROR.
 22   WRITE (6,96)
 96   FORMAT ( 58HOFOLLOWING CARDS WERE NOT PROCESSED DUE TO PREVIOUS ER
     1ROR.       )
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 25   READ (5,95) (KDATA(I),I=1,14)
      WRITE (6,906) (KDATA(I),I=1,14)
C  LOOK FOR END CARD FOR THIS VARSUM SET
      IF (KDATA(1) .EQ. IEND(1)) GOTO 20
      GOTO 25
C
C. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C     PROCESS VARSUM CARD
 21   WRITE(6,92) KDATA(1),ISID1,ISID2,IHEAD
 92   FORMAT(1H1,A6,1X,A6,A2,1X,I2,/)
      MID = JCOMM(6)
      IF (ISID1 .EQ. JCOMM(1) .AND.
     1    ISID2 .EQ. JCOMM(2) .AND.
     1    MID   .EQ. JCOMM(6))    GOTO 24
      WRITE (6,89)
 89   FORMAT (45HISID OR MID DOES NOT MATCH TAPE, JOB KILLED#     )
      GOTO 2000
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C READ PAGE HEADING CARDS
 24   IF (IHEAD .EQ. 0) GO TO 3083
C MAX 6 HEADER CARDS
      J2 = 14*IHEAD
      IF (IHEAD .LE. 6) GO TO 23
      WRITE (6, 85)
 85   FORMAT (28H1TOO MANY PAGE HEADING CARDS)
      GO TO 22
 23   READ (5,95) (IHEADC(J1), J1=1,J2)
 95   FORMAT (13A6,A2)
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C READ CONTROL CARD TYPE 2
3083  READ(5,94) IDREPT, ITOUT, ITITLE
 94   FORMAT (A6,1X,I1,1X,8A6,1X,I2)
      LINES = 55
      IF (IDREPT .NE. KBLANK) GO TO 26
      WRITE (6,84)
 84   FORMAT (25H1NO REPORT IDENTIFICATION      )
      GO TO 22
 26   KTOUT = 2 - ITOUT
      IF(KTOUT.LE.0) KTOUT = 1
      ITOUT = KTOUT - 1
      NGP = 0
      NSPEC = 0
      KT = 1
C KT IS SPEC CARD TABLE INDEX
C******
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C READ VXXX SPECIFICATION CARDS AND BUILD TABLE
 30   READ (5,93) KDATA(1), IV      , IVAR, LO, IHI,
     1         ISIZE, IFREQ, IMEAN, IMB, JOTHER, CLO, CHI, CSIZE,FACTOR
 93   FORMAT (A6,1X,A1, I3,1X,A6,1X,A6,1X, A6,1X, 4I1, 2X, 4F10.4  )
C LOOK FOR END CARD
      IF (KDATA(1).EQ.IEND(2)) GOTO 51
      IF(KDATA(1) .NE. KBLANK ) GOTO 36
C**** COMPUTE NUMBER DIFFERENT VARIABLES
      I = IVAR
      IF (IV .EQ. KWORDC) I=-1
C****  SEE IF THIS VARIABLE SAME AS PREVIOUS ONE.
      IF (I .NE. (KTABLE(KT-2)/MBIT))NSPEC=NSPEC+1
C   LOOK FOR ERRORS IN INTERVAL SPEC CARD
      IF (IFREQ+IMEAN+IMB+JOTHER .EQ. 0) GO TO 36
      IF (IVAR .LE. 0) GOTO 36
      IF (IV .EQ. KWORDC) GOTO 3000
      IF (LO .EQ. KBLANK) GO TO 36
      IF (IVAR .GT. NVARS) GOTO 36
      LENGTH = 0
      ICALC = 4
      IF (ISIZE .NE. KBLANK) GOTO 31
      ISIZE = 0
      IF (IMEAN .EQ. 0) GOTO 47
 45   WRITE (6,48)
      GOTO 36
 48   FORMAT (62 H1CANNOT CALCULATE MEAN OR STD DEV EXCEPT ON NUMERIC IN
     *TERVALS.)
 47   ICALC = 1
      IF (IHI .EQ. KBLANK) GOTO 33
C LOOK FOR BOTH LIMITS ALPHA J OR BEYOND
      ICALC = 2
      GOTO 33
 36   WRITE(6,8791)
8791  FORMAT('1 INVALID FIELD IN FOLLOWING CARD:')
      READ(0,95) (KDATA(I),I=1,14)
      WRITE(6,906) (KDATA(I),I=1,14)
 91   FORMAT (  1H1,A6,1X,A1, I3,1X,A6,1X,A6,1X, A6,1X, 4I1, 2X, 3F10.4
     1, 6X,     32HINVALID FIELD IN THIS SPEC CARD.)
      GO TO 22
 35   NS = ISPEC*15
      WRITE (6,37)  KTLAST,NS
 37   FORMAT (54H1MEMORY CAPACITY EXCEEDED. REDUCE NUMBER OF SPEC CARDS/
     *  1H ,
     * I6, 49H CELLS FOR VARIABLES ARE NEEDED IN THIS RUN PLUS
     * I6, 24H CELLS TOTAL FOR TITLES          )
      GOTO 22
C
C TEST FIELDS FOR COMPUTED VARIABLE
C FIRST 3 VARIABLES ARE INTEGERS, CANT BE COUNTED
3000 IF (IVAR .LT. 4 .OR. IVAR .GT. NCVAR) GOTO 36
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      IF(CSIZE .EQ. 0..AND. CHI .NE. 0..AND. CLO .GT. CHI)GOTO 36
      IF (CSIZE .LT. 0.)GOTO 36
      IF (CSIZE .NE. 0. .AND. CHI .LT. CLO) GOTO 36
      IF (FACTOR .EQ. 0.0) FACTOR = .0001
      CSAVE = CHI
       CHI = CLO + CSIZE
      ICALC = 3
      IF (CSIZE .NE. 0.0) GOTO 43
      ICALC = 2
      IF (CHI .NE. 0.0) GOTO 43
      ICALC = 1
 43   CTABLE(KT) = CLO
      CTABLE(KT+1) = CHI
      CTABLE(KT+2) = FACTOR
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C NEGATIVE VARIABLE NUMBER WILL INDICATE THIS IS A COMPUTED VARIABLE
      KTABLE (KT+3) =-IVAR*MBIT
      GOTO 40
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C SET SPEC INFO INTO KT TABLE
C EACH SPEC CARD CREATES A 5 WORD BLOCK OF IDENTIFYING INFO SUCH AS
C       ITS LIMITS, NUMBER, ETC.
C EACH INTERVAL CREATES A 3 WORD BLOCK FOR THE COUNTS.
C  THE INFO PORTION OF THE TABLE APPEARS ONLY ONCE, BUT THE
C COUNT BLOCKS ARE REPEATED ONCE FOR EACH REQUEST CARD.
C   10 REQUESTS WILL FIT ON ONE PRINT PAGE ACROSS.
C*******  FORMAT OF THE INFO BLOCK IS
C1 LO VALUE
C2 HI VALUE
C3 IF INTERGER GENERATED INTERVAL, THIS WORD CONTAINS THE DIVISOR
C         NEEDED TO SCALE OFF TRAILING BLANKS FROM THE DATA WORDS.
C3   IF A COMPUTED VARIABLE, THIS WORD CONTAINS THE FLOATING POINT
C         FACTOR WHICH DEFINES THE DISTANCE BEYOND THE END OF THE
C       INTERVAL (UPPER LIMIT) WHICH IS STILL TO BE INCLUDED IN THE
C       INTERVAL.  EXAMPLE..1-10 BY TWO GENERATES 5 INTERVALS, THE LAST
C         OF WHICH IS 9 TO 10+FACTOR.  THIS MAKES UPPER LIMIT INCLUSIVE.
C4VARIABLES REL LOC IN DATA, OBTAINED FROM IREL TABLE, ADDRESS PORTION
C   THIS FIELD ALSO CONTAINS VAR NUMBER IN DECREMENT
C         VARIABLE NUMBER IS + IF HISTORY, - IF COMPUTED.
C5 BITS 21-35 ARE PROCESSING INDICATORS.  BITS 3-17 IS TYPE CALCULATION.
C**** EACH COUNT BLOCK IS FROM 1 TO 3 WORDS, ONE FOR COUNT AND
C          TWO FOR MEAN AND STD DEV IF A NUMERIC INTERVAL.
C1 ACTUAL FREQUENCY COUNT
C2 SUM OF VALUES FOR MEAN CALCULATION
C3 SUM OF VALUES SQUARED
C
 31    DO 29 N = 1, 6
```

LINE #   TEXT...

```
      LENGTH = 7 - N
      CALL BCDBIN(   LO,  LLO,LENGTH,$29)
      CALL BCDBIN(  IHI, LIHI,LENGTH,$29)
      CALL BCDBIN(ISIZE,LISIZE,LENGTH,$29)
      GO TO 39
 29   CONTINUE
      GO TO 36
 39   ICALC = 3
      LO = LLO
      IHI = LIHI
      ISIZE = LISIZE
      ISAVE = IHI
      IHI = LO + ISIZE
 33   KTABLE(KT) = LO
      KTABLE(KT+1) = IHI
 49   KTABLE(KT+2) = LENGTH
      KTABLE(KT+3) = MOD(IREL(IVAR), MBIT) + IVAR*MBIT
 40   KTABLE(KT+4)=IFREQ+10*IMEAN+100*IMB+1000*JOTHER+ICALC*MBIT
 32   KT = KT + 5
      KTSIZE = KT-1
C ****KTSIZE IS LENGTH OF INFO ID BLOCK
C****     NGP IS GENERATED LENGTH OF COUNT TABLE
      NGP = NGP + 1
      IF (KT.GT.40000) GO TO 35
      IF (ISIZE .EQ. 0 .AND. CSIZE .EQ. 0.) GOTO 30
      NGP = NGP + 2
      IF (IV .EQ. KWORDC) GOTO 41
C
C
C SETUP MULTIPLE TABLE BLOCKS ON INTERVAL CARDS
      LO = LO + ISIZE
      IHI = LO + ISIZE
      IF (LO.LE.ISAVE) GO TO 33
CSET LAST HI TO ORIGINAL VALUE PLUS UNITY
      KTABLE(KT-4) = ISAVE + 1
      GO TO 30
C********
C SETUP MULTIVLE BLOCKS ON FLOATING POINT COMPUTED VARIABLES
 41   CLO = CLO + CSIZE
      CHI = CLO + CSIZE
      IF (CLO .LE. CSAVE) GOTO 43
      CTABLE(KT-4) = CSAVE*FACTOR
      GOTO 30
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C   BEGIN READING REQUEST CARDS
 51   NREQ = 1
      IF(END.EQ.1.0) GO TO 1
      IF (KT .NE. 1 .OR. KTSIZE .NE. 0) GOTO 52
```

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 58   WRITE (6,57)
      GOTO 22
 57   FORMAT (25HINO SPEC OR REQUEST CARDS      )
 52   READ (5,90) (REQCRD(I,R), I=1,10), KSEQ
 90   FORMAT (A4, 9A6, 18X, I4)
      REQCRD(11,R) = KSEQ
C  TEST FOR END CARD
      END = 1.
      IF (REQCRD(1,R) .EQ. IEND(1)) GO TO 60
      END = 0.0
 54   NREQ = NREQ+1
C  MAX 10 REQUESTS AT A TIME.
      IF (NREQ .EQ. 11)  GO TO 60
      GO TO 52
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C.....    CLEAR COUNT FIELDS AND SET UP COUNT BLOCK POINTERS
60    IF(R.EQ.1) GO TO 998
      IF(NGP.NE.0) KGP = NGP
      NREQ = NREQ-1
      IF (NREQ .LE. 0) GOTO 58
      KTLAST = KGP * NREQ + KTSIZE
      IF (KTLAST.GT.40000) GO TO 35
      J = KTSIZE + 1
      DO 103 I=J,KTLAST
 103  KTABLE(I) = 0
C ALL REQUEST CARDS READ. NOW GET NMASK FOR EACH TSEQ.
C EACH NMASK IS 58 WORDS LONG AND IS STORED WITH ITS REQUEST CARD
C.......     SEE IF DICT TITLES WILL FIT IN CORE
      IF (NSPEC .NE. 0) ISPEC = NSPEC
      I = ISPEC * 15 + KTLAST       +1
      IF (I.GT.40000) GO TO 35
      REWIND IT
      ISEQ = 0
      REWIND IS
C FIND LARGEST SEQ NUMBER REQUESTED
      DO 61 I=1,NREQ
 61   IF (REQCRD(11,I) .GT. ISEQ) ISEQ = REQCRD(11,I)
      MATCH = 0
       I= JCOMM(4)
       DO 70 J= 1,I
 70   READ(IS) K
 67   READ(IT,END=666) IDSP, K,(I,L=1,K), NC35, (NMASK(I),I=1,NC35),
     1 JSEQ, IKILLN,(IKILL(I),I=1,IKILLN)
      MI = IKILL(6)
      IF (JSEQ .LE. ISEQ)  GOTO 62
 666  CONTINUE
      WRITE (6,87)
 87   FORMAT (79HICANNOT FIND ALL TAPE SEQ NUMBERS FOR THIS VARSU
     .THIS SET IS SKIPPED.                     )
      GOTO 51
```

```
C SEARCH REQ CARDS FOR CORRESPONTING SEQ NUMBER TO TAPE REC JUST READ
  62   DO 63 I=1,NREQ
       IF (JSEQ .NE. REQCRD(11,I)) GOTO 63
       MATCH = MATCH+1
       IKILL1 = IKILL(35)
       IKILL2 = IKILL(50)
       IF(IKILL1.EQ.'JOBGRP'.AND.IKILL2.NE.'FIXBJD') GO TO 3333
       IF(REQCRD(2,I).NE.' ') GO TO 6668
       REQCRD(2,I) = IDSP
       IF(REQCRD(3,I).EQ.' ') ENCODE(6669,REQCRD(3,I))(IKILL(J),J=37,44)
 6669  FORMAT(8A6)
 6668  IF(MID.EQ.MI) GO TO 68
       WRITE(6,38) MID, MI, IDSP
 38    FORMAT(1H1,'MATRIX ID MISMATCH ',A6,' NOT = ',A6,' OF ',A6,/,
      1' THIS SET IS SKIPPED')
       GO TO 51
 68    CONTINUE
       N58 = NC35
       IF(NC35.LE.58) GO TO 64
       WRITE(IS) I,        (NMASK(K),K=1,NC35)
       N58 = 58
C SEQ MATCH, PUT NMASK IN WITH REQ CARD
 64    DO 65 J = 1, N58
 65    REQCRD(J+11,I) = NMASK (J)
  63   CONTINUE
       IF (MATCH .LT. NREQ) GOTO 67
 69    REWIND IT
       MSKWRD = N58 + 11
       REWIND IS
C
       REWIND IP
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
       DO 102  I=1,10
 102   MBRS(I) = 0
C SKIP OVER COMM REGION AND REL TABLE
       READ(IP) K
       READ(IP) K
C SKIP OVER DICTIONARY RECS ON HISTORY DATA TAPE
       II = JCOMM(4)
       DO 10 I=1,II
 10    READ(IP) K
C
C • • • • • • • • • • • • • • • • • •
C GATHER THE DATA AS FOLLOWS.  CHECK A POSITION IN EACH NMASK.
C EACH REQ CARD HAS ITS NMASK WITH IT. WHEN A GIVEN POSITION OF ANY
C NMASK SHOWS THAT THIS DATA RECORD SHOULD BE COUNTED, THEN SPACE
C DOWN THE HISTORY TAPE AND READ THAT RECORD INTO KDATA.
C SWITCH NOREAD TELLS WHETHER TO READ THE TAPE OR WHETHER THE
C DESIRED RECORD IS ALREAD IN KDATA FOR SUCCESSIVE REQ CARDS.
C SWITCH LASTR KEEPS TRACK OF HOW MANY RECORDS WE ARE DOWN THE TAPE.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
C THE DATA IN KDATA IS THEN MATCHED AGAINST THE VARIABLES REQUESTED
C IN KTABLE.
C AS SOON AS THAT KDATA HAS BENN PROCESSED AGAINST ALL KTABLE BLOCKS,
C THEN THE NEXT POSITION IN NMASK IS INSPECTED TO SEE IF THE NEXT
C TAPE RECORD IS TO BE COUNTED. THIS IS CONTINUED FOR ALL POSITIONS
C IN NMASK FOR EACH REQUEST CARD, UP TO 10 REQUEST CARDS AT A TIME.
C SUM ONLY THOSE WHICH HAVE NMASK BIT SET
      NCASE = 1
      LASTR =0
      NVSTRT = NCVAR+1
      NVEND = NVSTRT + NVCNT- 1
 2003 IF(NC35.LE.58) GO TO 2005
       IF(NCASE .EQ. 1) GO TO 2005
C.... SKIP DICT TITLES ON WORK TAPE
      II = JCOMM(4)
      DO 2007 I=1,II
 2007 READ(IS) K
      I = (NCASE/2030)*58
      DO 2006 K=1,NREQ
      READ(IS) N, (NMASK(L),L=1,NC35)
      DO 2002 M = 1, N58
      J=M+1
 2002 REQCRD(M+11,N) = NMASK(J)
 2006 CONTINUE
      REWIND IS
 2005 LIMIT = N58 + 11
      DO 500 NCOUNT = 12, LIMIT
      DO 500 NBIT = 1,35
      NOREAD = 0
      DO 499 KB = 1,NREQ
      IF (AND(AEQCRD(NCOUNT,KB),RMASK(NBIT)) .EQ. 0.0) GOTO 499
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C THIS CASE NEEDS TO BE COUNTED
      MBRS(KB) = MBRS(KB) + 1
C COMPUTE INDEX IN KTABLE FOR THIS REQUEST CARD
      JS = KGP * (KB-1) + KTSIZE + 1
C GET DATA FROM HISTORY TAPE UNLESS ALREADY IN CORE (NOREAD = 1).
      IF (NOREAD .NE. 0) GOTO 105
      I = NCASE - LASTR
      NOREAD = 1
      LASTR = NCASE
      DO 106 J=1,I
  106 READ(IP,ERR=896) K,(KDATA(KA),KA=1,NCVAR),K,(KDATA(KA),KA=NVSTRT,
     1 NVEND)
      GO TO 897
  896 WRITE(6,898) (KDATA(NZ),NZ=1,5), NCOUNT, JCOMM(3)
  898 FORMAT(' READ ERROR ON HIS/KPATH FILE.',/,1X,5A6,2X,I6,2X,I6)
      CALL ERTRAN(2)
  897 CONTINUE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
  105  DO 401 KT=1,KTSIZE, 5
C*******
C CALC CODES ARE AS FOLLOWS
C    FOR COMPUTED VARIABLES, FLOATING POINT VALUES
C           1= EQUAL COMPARE TO LOW LIMIT ONLY
C           2= VALUE WITHIN INTERVAL LOW TO HIGH INCLUDING END POINTS
C           3= GENERATED INTERVAL, VALUE GE LOW BUT LT HIGH
C    FOR HISTORY VARIABLES
C           1= VALUE EQUAL TO LOW LIMIT ONLY
C           2= POSITIVE RANGE INCLUDING END POINTS
C           3= SAME AS CALC 4 BUT VALUE IS 6 CHARACTERS LONG.
C           4=GENERATED INTERVAL, VALUE GE LOW BUT LT HIGH
C                CODES 3 AND 4 REQUIRE DATA TO BE CONVERTED TO BINARY.
C           5= ALPHA INTERVAL LOW + BUT HIGH -  , SUCH AS A TO J
C           6=ALPHA INTERVAL LOW AND HIGH BOTH MINUS, SUCH AS J TO R
C
      ICALC = KTABLE(KT+4)/MBIT
      IF (KTABLE(KT+3) .LT. 0) GOTO 200
C GET VARIABLES LOC IN DATA FIELD
      KLOC = MOD(KTABLE(KT+3), MBIT) + NCVAR
      IDATA = KDATA(KLOC)
      GO TO (305,307,110,110,307,307), ICALC
  305 IF(IDATA.NE.KTABLE(KT)) GO TO 400
      GO TO 310
  307 JUMP = LASCMP(IDATA,KTABLE(KT),KTABLE(KT+1),1) + 1
      GO TO (400, 310), JUMP
  110 CALL BCDBIN(IDATA,IDATA,KTABLE(KT+2),$400)
      IF(IDATA.LT.KTABLE(KT).OR .IDATA.GE.KTABLE(KT+1)) GO TO 400
  310 KTABLE(JS) = KTABLE(JS) + 1
      GOTO (400,400,306,306,400,400), ICALC
  306 IF (MOD(KTABLE(KT+4),100) .LT. 2) GOTO 400
C SUM X AND XSQR FOR MEAN AND SD
  303 IF (IDATA .EQ. 0) GOTO 400
      CDATA = IDATA
      GOTO 212
C COUNT COMPUTED VARIABLES WHICH ARE FLOATING POINT
  200 KLOC =-KTABLE(KT+3)/MBIT
      CDATA = DATA(KLOC)
      GOTO (201,202,210), ICALC
  201 IF (CDATA .EQ. CTABLE(KT))   GOTO 209
      GOTO 400
  202 IF (CDATA .LT. CTABLE(KT) .OR. CDATA .GT. CTABLE(KT+1)) GOTO 400
      GOTO 209
C DO A RANGE COMPARE
  210 IF (CDATA .LT. CTABLE(KT) .OR. CDATA .GE. CTABLE(KT+1))GOTO 400
  209           KTABLE(JS  ) = KTABLE(JS  ) + 1
C SUM X AND XSQR
  211 IF (CDATA .EQ. 0.0) GOTO 400
      IF (MOD(KTABLE(KT+4),100) .LT. 2) GOTO 400
  212 CTABLE(JS+1) = CTABLE(JS+1) + CDATA
```

LINE # TEXT...

```
      CTABLE(JS+2) = CTABLE(JS+2) + CDATA*CDATA
 400  GOTO (401,401,402,402,401,401), ICALC
 402  JS = JS+2
 401  JS = JS+1
 499  CONTINUE
      NCASE = NCASE + 1
      IF (NCASE .GT. JCOMM(3)) GOTO 2004
 500  CONTINUE
      GOTO 2003
C
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C READ VARIABLE TITLES, CALLED THE DICTIONARY
C WORDS 14 AND 15 CONTAIN FORTRAN FORMAT FOR TITLE PRINTING
2004  REWIND IP
      REWIND IS
C FIND DICTIONARY TITLES TO MATCH VARIABLES REQUESTED
C AFTER A TITLE IS OBTAINED FROM TAPE, IT IS SAVED IN CORE FOLLOWING THE
C KTABLE.
       NCT = NCB
       IF (IDREAD.EQ.1) GO TO 18
       IDREAD = 1
       NCC = 0
      L = KTLAST
      IDSTRT = L+1
      JC = JCOMM(4)
       ID = NCT
      DO 15 II= 1,JC
 11   READ(IS) K, ((KDICT(J,M), J=1,15), M=1,K)
      NCC = NCC + K
      DO 17 M=1,K
      DO 12 I=1,KTSIZE,5
      IF (KTABLE(I+3)/MBIT .EQ. ID) GO TO 13
 12   CONTINUE
      GO TO 17
 13   DO 14 J=1,15
      IB = L + J
 14   KTABLE (IB) = KDICT (J ,M)
      KTABLE (IB) = ID
      L = L + 15
      IF (L.GT.40000) GO TO 35
 17    ID = ID + NCT
      IF (NCC.NE. NCVAR) GOTO 15
C WHEN COMPUTED TITLES FINISHED, GO POSITIVE FOR HISTORY TITLES
       NCT = 1
      ID = 1
 15   CONTINUE
      IDSIZE = L
      REWIND IP
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C   ALL COUNTS FINISHED
CPRINT PAGE HEADINGS
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
 18    NLINES = 2
       NPAGE = NPAGE +1
       DO 598 I=1,KTOUT
       IOUT = KOUT(I)
C
551    WRITE(IOUT,904) ITITLE, IDREPT, NPAGE
904    FORMAT(1H1,8A6,29X,A6,' PAGE ',I4,/)
C
       IF (IHEAD .EQ. 0) GOTO 560
 555 WRITE(IOUT,906)(IHEADC(K), K=1,J2)
906  FORMAT (1H , 14A6)
560  WRITE (IOUT, 875)
875  FORMAT (1H0,  5X,
    *          46HREPORTS ON THE FOLLOWING GROUPS WERE REQUESTED        )
      DO 561KB=1,NREQ
561    WRITE (IOUT,907) (REWCRD(K,KB), K=2,10), MBRS(KB)
907   FORMAT (1H ,  8X, 18HGROUP IDENTITY =                  ,
     1                                A6,2X,8A6,  1X,
     2     10HCONTAINING. I6, 9H MEMBERS.                  )
598   CONTINUE
      NLINES = IHEAD+9+NREQ
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      JSTART = KTSIZE + 1
      DO 701 KT=1,KTSIZE,5
C IF VAR NUMBER IS SAME AS LAST ONE, THIS IS SUCCESSIVE INTERVAL
 611  IF (KTABLE(KT-2)   .EQ. KTABLE(KT+3)) GOTO 621
C
C CLEAR PREVIOUS SUMMATION BOXES
      DO 616 I=1,10
      SUMX(I) = 0
      SUMX2(I) =0
      IOTHER(I) =0
      SD(I) = 0
      FMEAN(I) =0
 616  KOTHER(I) =0
      IVAR = KTABLE(KT+3)/MBIT
C*******
C SET UP PRINT SWITCHES
      J = MOD(KTABLE(KT+4) , MBIT)
      IFREQ = MOD(J,10)
      IMEAN = MOD(J,100) - IFREQ
      IMB = MOD(J, 1000) -    IMEAN - IFREQ
      JOTHER = 0
      IF (J .GE. 1000) JOTHER = 1
```

LINE # TEXT...

```
      JUMP = 1
C
C SEE IF ROOM LEFT ON PAGE FOR THIS BLOCK OF PRINT
      ILINES = 4
      IF (IFREQ .EQ. 0) GOTO 602
      ILINES = ILINES + 1
      DO 601 KI = KT, KTSIZE, 5
      IF (KTABLE(KI+3) .NE. KTABLE(KI+8 ))GOTO 602
 601  ILINES = ILINES + 1
 602  IF (JOTHER .NE. 0) ILINES = ILINES +1
      IF (IMB .NE. 0) ILINES = ILINES + 1
      IF (IMEAN . NE. 0) ILINES = ILINES + 2
      IF(ILINES+NLINES .GT. LINES .AND. ILINES .LT. LINES) GOTO 754
C
C RETRIEVE DICTIONARY FOR THIS TITLE
C THESE ARE STACKED IN 15 WORD BLOCKS
C
 604  J = IDSTRT
      DO 605 JD = J, IDSIZE, 15
      IF (KTABLE(JD+14) .EQ. IVAR) GOTO 606
 605  CONTINUE
      STOP
 606  K = JD + 13
      DO 610 I=1,KTOUT
      IOUT = KOUT(I)
 610  WRITE(IOUT,900)       (KTABLE(N),N=JD,K)    , (REQCRD(2,L),L=1,R)
 900  FORMAT(/18HOVARIABLE NUMBER=     , 13A6, A2, /
     1  11H   INTERVAL, 9X, 10(5X, A6)    )
      NLINES = NLINES+4
C
C ADD UP MEMBERS COUNTED
 621  ICALC = KTABLE(KT+4)/MBIT
      J = JSTART - KGP
      DO 670 IO = 1,M
      J = J + KGP
 670  KOTHER(IO) = KOTHER(IO) + KTABLE(J)
C
C DID HE WANT FREQ COUNTS PRINTED
      IF (IFREQ .EQ. 0) GOTO 659
C
C IVAR FIELD NEGATIVE IF COMPUTED VARIABLE
      IF(IVAR .LT. 0) GOTO 3650
      GOTO (623,623,650,650,623,623), ICALC
C****** PRINT LINES WITH BCD LIMITS
 623  DO 622 I=1,KTOUT
      IOUT = KOUT(I)
 622  WRITE (IOUT    ,901) KTABLE(KT), KTABLE(KT+1),(KTABLE(J),
     *   J=JSTART,KTLAST,KGP )
 901  FORMAT (1H , A6,1X, A6, 5X, 10I11)
      GOTO 630
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C PRINT COMPUTED COUNTS WHERE INTERVAL IS FLOATING POINT
 3650  DO 3653 I=1,KTOUT
       IOUT = KOUT(I)
       GOTO (3651,3654,3652), ICALC
 3651 WRITE (IOUT, 3910) CTABLE(KT), (KTABLE(J),J=JSTART,KTLAST,KGP)
 3910 FORMAT (1H , F9.4, 9X, 10I11)
       GOTO 3653
 3652  CTEMP = CTABLE(KT+1) - .0001
       XTEMP = CTABLE(KT) + CTABLE(KT+2) - CTABLE(KT+1)
       IF(XTEMP.EQ.0.0) CTEMP = CTABLE(KT+1)
 3655 WRITE (IOUT, 3909) CTABLE(KT),CTEMP,(KTABLE(J),J=JSTART,KTLAST,KGP
     .  )
 3909 FORMAT (1H , 2F9.4, 10I11)
       GOTO 3653
 3654 CTEMP = CTABLE(KT+1)
 3653 CONTINUE
       GOTO 630
C PRINT NUMERIC INTERVALS
  650   DO 653  I = 1,KTOUT
       IOUT = KOUT(I)
        KTEMP = KTABLE(KT+1) -1
        IF (KTABLE(KT).NE.KTEMP) GO TO 652
       WRITE (IOUT   ,910) KTABLE(KT),(KTABLE(J),J=JSTART,KTLAST,KGP)
  910   FORMAT (1H , I6, 12X, 10I11)
       GOTO 653
  652  WRITE (IOUT   ,909) KTABLE(KT), KTEMP, (KTABLE(J),J=JSTART,
     .      KTLAST,KGP)
  909  FORMAT (1H , I6,1X, I6, 5X, 10I11)
 653   CONTINUE
  630   NLINES = NLINES+1
  659  IF (IMEAN .EQ. 0) GOTO 661
       J=JSTART-KGP
       DO 660 IO = 1,NREQ
       J = J + KGP
       SUMX(IO) =  SUMX(IO) + CTABLE(J+1)
  660  SUMX2(IO) = SUMX2(IO) + CTABLE(J+2)
C
  661  JUMP = 2
       GOTO 750
C IF NEXT VARIABLE SAME AS THIS LAST ONE, DONT PRINT MEAN OR SD YET
C FORCE THIS PATH ON LAST VARIABLE IN TABLE
  631  IF (KTABLE(KT+3) .EQ. KTABLE(KT+8) .AND. KT .NE.(KTSIZE-4))GOTO700
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C WRITE OTHER COUNT
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C PRINT NUMBER OF MEMBERS ACTUALLY COUNTED, EXCLUDING OTHER
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: MAIN PROGRAM PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
 634  IF (IMB .EQ. 0) GOTO 645
      DO 643 I=1,KTOUT
      IOUT = KOUT(I)
 643  WRITE (IOUT, 644) (KOTHER(J), J=1,NREQ)
 644  FORMAT (14H TOTAL COUNTED , 5X, I11, 9(1H*, I10), 1H*  )
      NLINES = NLINES + 1
 645  DO 633  I=1,NREQ
 633  IOTHER(I)  =  IABS(MBRS(I) - KOTHER(I))
      IF (JOTHER .EQ. 0) GOTO 641
      DO 632 I=1,KTOUT
      IOUT = KOUT(I)
 632  WRITE (IOUT   , 902) (IOTHER(J), J= 1,NREQ)
 902  FORMAT (1H , 8X, 5HOTHER, 5X, 10I11)
       NLINES = NLINES+1
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C TEST IF MEAN AND SD NEED PRINTING
 641  IF (IMEAN .EQ. 0) GOTO 700
      DO 680 IO = 1,NREQ
      TMBRS(IO) = KOTHER(IO)
      FMEAN(IO) = SUMX(IO)/TMBRS(IO)
 680  SD(IO) = SQRT ( SUMX2(IO)/TMBRS(IO) - (FMEAN(IO)**2))
      DO 642 I=1,KTOUT
      IOUT = KOUT(I)
      WRITE (IOUT   ,903) (FMEAN(J), J=1,NREQ)
 903  FORMAT (1H , 9X, 4HMEAN, 5X, 10F11.4)
 642  WRITE (IOUT   ,922) (SD(J), J=1,NREQ)
 922  FORMAT (1H , 13HSTD DEVIATION, 5X, 10F11.4 /)
       NLINES = NLINES+3
 700  GOTO (701,701,702,702,701,701), ICALC
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C PRINT PAGE NUMBER
  750 IF (NLINES .LT. LINES) GOTO 752
 754  NPAGE = NPAGE + 1
      DO 751 I=1,KTOUT
      IOUT = KOUT(I)
      IF (ILINES .LT. LINES) GOTO 751
      WRITE (IOUT   , 753)
 753  FORMAT (2H0, 80X, 23HCONTINUED ON NEXT PAGE          )
 751  WRITE(IOUT,904) ITITLE, IDREPT, NPAGE
      NLINES = 2
 752  GOTO (604,631), JUMP
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
 702   JSTART = JSTART + 2
```

LINE # TEXT...

```
 701   JSTART = JSTART + 1
       NREPT = 1
       IF(END.EQ.1.0) GO TO 998
       GO TO 51
 998   IF(ITOUT.EQ.1) CALL REPEND(ISID1,IDREPT,ITITLE,LSEQ)
       GO TO 1
C
C
C
C ALL REPORTS COMPLETED. CLOSE OUTOUT TAPE
C
C PREPARE TO ABORT
C NREPT INDICATES WHETHER ANY REPORTS WERE WRITTEN ON C4
 2000  IF(NREPT.EQ.0) CALL ERTRAN(2)
       IF(ITOUT.EQ.1) CALL REPEND(ISID1,IDREPT,ITITLE,LSEQ)
 1000  REWIND IP
       REWIND IT
       REWIND IS
       CALL MARGIN
       STOP
 3333  CONTINUE
       WRITE(6,3334) JSEQ
 3334  FORMAT(' RUN FIXBJD ON THIS BJD FILE FOR REPORT',I6,
      1 /,'0RUN ABORTS DUE TO BAD JOBGRP MASK ARRAY.')
       CALL ERTRAN(2)
       END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

THIS IS A LIST OF THE ENTRY POINTS INTO THE CODAP SUBROUTINE LIBRARY. THE LEFT-HAND COLUMN IS AN ALPHABETIZED LIST OF THE ENTRY POINTS, AND IN THE RIGHT-HAND COLUMN IS THE NAME OF THE ELEMENT IN WHICH THE ENTRY POINT MAY BE FOUND.

ASGA	ASGA
ASGAM	ASGAM
ASGC	ASGC
ASGCM	ASGCM
ASGCMR	ASGCM
AUTORV	AUTORV
BCDBIN	DATACV
BIN	SUBRTN
BINBCD	DATACV
BINFD	SUBRTN
BINOCT	SUBRTN
BLANK	BLANK
CARD	SUBRTN
CHKMSK	MSKOPI
CLOSEF	CFHIO
CLOSER	RFILE
COMPA	COMPA
COMPC	COMPC
COMPN	COMPN
COMPRC	COMPRC
CPI	CPI
CYCLES	CYCLES
DATE	SUBRTN
DATETIME	SUBRTN
DATETM	DATETM
DEFINE	RFILE
DELAY	DELAY
DISK	DISK
DISTX1	DISTX1
DKSTAT	DKSTAT
DREAD	DREAD
DSKRED	DISK
DSKSTO	DISK
ERII	ERII
ERIIX	ERII
ERRMSG	ROSTER
ERSMSK	MSKOPI
ERTRAN	ERTRAN
FACREJ	FACREJ
FDBIN	SUBRTN
FIELD	SUBRTN
FLD	FLD
FMTGEN	FMTGEN
FORMAT	FORMAT
FORM2X	FORM2X

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

FREDEV	FREDEV
FREE	FREE
FREES	FREES
FSORT	FSORT
FSORT2	FSORT
GETC	SUBRTN
GETFIL	GETFIL
GETMAS	GETMAS
GETMSK	MSKOP2
GETMSM	MSKOP2
GETPCD	GETPCD
GETPCT	GETPCT
GETRAN	IRANF
H2ACND	H2ACND
H2SORT	H2SORT
INFO	SUBRTN
INSERT	INSERT
IRANF	IRANF
INTIAL	INTIAL
IRAND	RANDOM
ISCAN	SUBRTN
JDCOPY	JDCOPY
JDEOF	JDEOF
LASCMP	LASCMP
LISORT	LISORT
LSORT	LSORT
MARGIN	NOHEAD
MARGNC	NOHEAD
MOVBIN	SUBRTN
MOVFD	SUBRTN
MOVFD1	SUBRTN
NEXREL	NEXREL
NEXREM	NEXREL
NOHEAD	NOHEAD
NOPAGE	NOHEAD
NRAND	NRAND
OCTBIN	SUBRTN
OMSG	OMSG
OMSGW	OMSG
OMSGWO	OMSGO
OMSGO	OMSGO
OPENF	CFHIO
OVRFLO	OVRFLO
PARTBL	SISO
PRINTC	ROSTER
PRTDIC	PRTDIC
PUTC	SUBRTN
RANF	RANF
RANSEQ	RANSEQ
READF	CFHIO

412

READR	RFILE
REPEND	REPEND
REPEN1	REPEN1
RESDRV	RESDRV
RESET	ROSTER
REWIND	TREAD
REWINM	DREAD
RPCOPY	RPCOPY
RPEOF	RPEOF
RPINDX	RPINDX
RUNID	RUNID
SAMPLE	SAMSEL
SBUILD	LINK
SCAN	SUBRTN
SELECT	SAMSEL
SETMSK	MSKOP1
SETSCN	SUBRTN
SETUP	SETUP
SETUP5	SETUP5
SETUP9	SETUP9
SHIFT	SHIFT
SICLS	SISO
SIGET	SISO
SIOPN	SISO
SOCLS	SISO
SOOPN	SISO
SOPEN3	TABL3
SOPUT	SISO
SPOST	SISO
SPREP	SISO
SHAND	RANDOM
SHREL	LINK
SHRET	LINK
SSORT	LINK
STATS	STATS
STATSS	STATS
STATUS	STATUS
TIME	SUBRTN
TMTCSC	TMTCSC
TREAD	TREAD
USEMSK	MSKOP2
WRITEF	CFHIO
WRITER	RFILE
ZERBLK	ZERBLK
ZERMSK	MSKOP1
ZERO	ZERO
ZERO1	ZERO1
ZERO	SUBRTN

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
      SUBROUTINE ASGA(NAME,IUSE)
      DIMENSION NAME(3), NAM(2), ID(2), IMAGE(8), IFREE(3)
      DATA IBLANK /' '/
      ENCODE(200,IFREE) IUSE
 200  FORMAT('@FREE,B ',I2,'. . ')
      WRITE(6,177) IFREE
      CALL ERTRAN(6,IFREE)
      ENCODE(201,ID) IUSE, IBLANK
 201  FORMAT(I2,4X,A6)
      NAM(1) = NAME(1)
      NAM(2) = NAME(2)
      IF(NAME(3).NE.IBLANK) GO TO 50
      ENCODE(100,IMAGE) IUSE, NAM
 100  FORMAT('@USE ',I2,',',A6,A4,'(+1) . ')
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(ID,ID1,ID2,IFLAG)
      IF(IFLAG.EQ.0) GO TO 900
      CALL DKSTAT(ID,ID1,ID2,IFLAG)
      IF(IFLAG.EQ.0) GO TO 900
      CALL ERTRAN(6,IFREE)
      ENCODE(101,IMAGE) IUSE, NAM
 101  FORMAT('@USE ',I2,',',A6,A4,' . ')
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(ID,ID1,ID2,IFLAG)
      IF(IFLAG.EQ.0) GO TO 900
      CALL DKSTAT(ID,ID1,ID2,IFLAG)
      IF(IFLAG.EQ.0) GO TO 900
      ENCODE(102,IMAGE) IUSE
 102  FORMAT('@FREE,R ',I2,' . ')
      CALL ERTRAN(6,IMAGE)
      ENCODE(103,IMAGE) NAM
 103  FORMAT('@ASG,YZ ',A6,A4,' . ')
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(ID,ID1,ID2,IFLAG)
      IF(IFLAG.EQ.1) GO TO 977
 905  ENCODE(102,IMAGE) IUSE
      CALL ERTRAN(6,IMAGE)
      ENCODE(104,IMAGE) NAM
 104  FORMAT('@ASG,AZ ',A6,A4,' . ')
      WRITE(6,177) (IMAGE(I),I=1,4)
 177  FORMAT(1X,8A6)
      CALL GETFIL(IMAGE)
      ENCODE(101,IMAGE) IUSE, NAM
      WRITE(6,177) (IMAGE(I),I=1,4)
      RETURN
 977  CONTINUE
      CALL DKSTAT(ID,ID1,ID2,IFLAG)
      IF(IFLAG.EQ.0) GO TO 905
      WRITE(6,105) NAM, ID1, ID2, IFLAG
 105  FORMAT(' FILE ',A6,A4,' IS NOT CATALOGUED IN THE MASTER FILE',
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
           1 ' DIRECTORY. ',/,' EQUIP = ',O6,' MODE = ',
           2 O6,' IFLAG = ',O6,' (ALL OCTAL CODES)')
            CALL ERTRAN(2)
   50       CONTINUE
            ENCODE(110,IMAGE) IUSE, NAME
  110       FORMAT('@USE ',I2,',',A6,A4,'(-',A1,') . ')
            CALL ERTRAN(6,IMAGE)
            CALL STATUS(ID,ID1,ID2,IFLAG)
            IF(IFLAG.EQ.0) GO TO 900
            CALL DKSTAT(ID,ID1,ID2,IFLAG)
            IF(IFLAG.EQ.0) GO TO 900
                CALL ERTRAN(6,IFREE)
            ENCODE(113,IMAGE) NAME
  113       FORMAT('@ASG,YZ ',A6,A4,'(-',A1,') . ')
            CALL ERTRAN(6,IMAGE)
                ENCODE(110,IMAGE) IUSE,NAME
                CALL ERTRAN(6,IMAGE)
            CALL STATUS(ID,ID1,ID2,IFLAG)
            IF(IFLAG.EQ.1) GO TO 978
  903       ENCODE(102,IMAGE) IUSE
            CALL ERTRAN(6,IMAGE)
            ENCODE(114,IMAGE) NAME
  114       FORMAT('@ASG,AZ ',A6,A4,'(-',A1,') . ')
            WRITE(6,177) (IMAGE(I),I=1,5)
            CALL GETFIL(IMAGE)
            ENCODE(110,IMAGE) IUSE, NAME
            WRITE(6,177) (IMAGE(I),I=1,5)
            RETURN
  978       CONTINUE
            CALL DKSTAT(ID,ID1,ID2,IFLAG)
            IF(IFLAG.EQ.1) GO TO 903
            WRITE(6,115) NAME, ID1, ID2, IFLAG
  115       FORMAT(' FILE ',A6,A4,'(-',A1,') IS NOT CATALOGUED IN THE',
           1 ' MASTER FILE DIRECTORY.',/,' EQUIP = ',O6,' MODE = ',O6,
           2 ' IFLAG = ',O6)
            CALL ERTRAN(2)
  900       CONTINUE
            WRITE(6,166) NAM, IUSE
  166       FORMAT(' FILE ',A6,A4,' ALREADY ASSIGNED.  USE = UNIT ',I2)
            RETURN
            END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE ASGAM(NAME,IUSE)
      DIMENSION NAME(3), NAM(2), ID(2), IMAGE(8), IFREE(3)
      DATA IBLANK /' '/
      ENCODE(200,IFREE) IUSE
200   FORMAT('@FREE,A ',I2,'. . ')
      WRITE(6,106) IFREE
      CALL ERTRAN(6,IFREE)
      ENCODE(201,ID) IUSE, IBLANK
201   FORMAT(I2,4X,A6)
      NAM(1) = NAME(1)
      NAM(2) = NAME(2)
      IF(NAME(3).EQ.IBLANK) NAME(3) = '0'
      CALL BCDBIN(NAME(3),IUNIT,1,$988)
      IF(IUNIT.GT.3) GO TO 988
      ENCODE(110,IMAGE) IUSE, NAME
110   FORMAT('@USE ',I2,',',A6,A4,'M(-',A1,') . ')
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(ID,ID1,ID2,IFLAG)
      IF(IFLAG.EQ.0) GO TO 900
      CALL DKSTAT(ID,ICODE1,ICODE2,IFLAG)
      IF(IFLAG.EQ.0) GO TO 900
      ENCODE(102,IMAGE) IUSE
      CALL ERTRAN(6,IMAGE)
102   FORMAT('@FREE,R ',I2,'. . ')
      ENCODE(113,IMAGE) NAME
113   FORMAT('@ASG,YZ ',A6,A4,'M(-',A1,') . ')
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(ID,ID1,ID2,IFLAG)
      IF(IFLAG.EQ.1) GO TO 978
905   ENCODE(102,IMAGE) IUSE
      CALL ERTRAN(6,IMAGE)
      ENCODE(114,IMAGE) NAME
114   FORMAT('@ASG,AZ ',A6,A4,'M(-',A1,'). . ')
      WRITE(6,106) (IMAGE(I),I=1,5)
106   FORMAT(1X,8A6)
      CALL GETFIL(IMAGE)
      ENCODE(110,IMAGE) IUSE, NAME
      WRITE(6,106) (IMAGE(I),I=1,5)
      RETURN
978   CONTINUE
      CALL DKSTAT(ID,ICODE1,ICODE2,IFLAG)
      IF(IFLAG.EQ.0) GO TO 905
      WRITE(6,115) NAME
115   FORMAT(' FILE ',A6,A4,'M(-',A1,') IS NOT CATALOGUED IN THE',
     1 ' MASTER FILE DIRECTORY.  PLEASE VERIFY FILENAME.')
      CALL ERTRAN(2)
988   CONTINUE
      WRITE(6,125) NAME
125   FORMAT(1X,A6,A4,'M(-',A1,') IS NOT A VALID CYCLE.')
      CALL ERTRAN(2)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

ASGAM   ASSIGN A CYCLE OF A 3-REEL FILE

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
 900    CONTINUE
        WRITE(6,166) NAM, IUSE
 166    FORMAT(' FILE ',A6,A4,'M ALREADY ASSIGNED.  USE = UNIT ',I2)
        RETURN
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE ASGC(NAME,IUSE)
      DIMENSION NAME(2), ID(2), IMAGE(14), IFREE(4)
      ENCODE(100,IMAGE) IUSE
100   FORMAT('@FREE,A ',I2,' . ')
      WRITE(6,106) (IMAGE(I),I=1,2)
      CALL ERTRAN(6,IMAGE)
      ENCODE(101,IFREE) IUSE
101   FORMAT('@FREE   ',I2,' . ')
      ENCODE(102,ID) IUSE
102   FORMAT(I2,10X)
      ISW = 0
1     CONTINUE
      ENCODE(103,IMAGE) NAME
103   FORMAT('@ASG,YZ ',A6,A4,' . ')
      CALL ERTRAN(6,IMAGE)
      ENCODE(104,IMAGE) IUSE, NAME
104   FORMAT('@USE ',I2,',',A6,A4,' . ')
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(ID,ID1,ID2,IFLAG)
      CALL DKSTAT(ID,ICODE1,ICODE2,JFLAG)
      CALL ERTRAN(6,IFREE)
      IF(IFLAG.EQ.0) GO TO 988
      IF(JFLAG.EQ.0) GO TO 988
      ENCODE(105,IMAGE) NAME
105   FORMAT('@ASG,UZ ',A6,A4,'(+1).,16N . ')
      WRITE(6,106) (IMAGE(I),I=1,5)
106   FORMAT(1X,14A6)
      CALL GETFIL(IMAGE)
      ENCODE(107,IMAGE) IUSE, NAME
107   FORMAT('@USE ',I2,',',A6,A4,'(+1) . ')
      WRITE(6,106) (IMAGE(I),I=1,5)
      CALL ERTRAN(6,IMAGE)
      RETURN
988   WRITE(6,108) NAME
108   FORMAT(' FILE ',A6,A4,' IS ALREADY CATALOGUED.')
      CALL BCDBIN(NAME(2),NUM,4,$989)
      NUM = NUM + 1
      ISW = ISW + 1
      IF(ISW.LT.6) GO TO 2
      WRITE(6,110)
110   FORMAT(' RUN ABORTS AFTER 5 NAME CHANGE ATTEMPTS.')
      CALL ERTRAN(2)
2     CALL BINBCD(NUM,NAME(2),4,$989,0)
      WRITE(6,111) NAME
111   FORMAT(' CHANGING FILE NAME TO ',A6,A4,/)
      GO TO 1
989   WRITE(6,112) NAME
112   FORMAT(' UNABLE TO ALTER FILE NAME ',A6,A4,'.  RUN HALTS')
      CALL ERTRAN(2)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
          SUBROUTINE ASGCM(NAME,IUSE)
          DIMENSION NAME(2), ID(2), IMAGE(14), IFREE(4), IFMT(6)
          DATA IBLANK/' '/
          DATA IFMT/'(A1)  (1XA1)(2XA1)(3XA1)(4XA1)(5XA1)'/
          ENCODE(100,IMAGE) IUSE
    100   FORMAT('@FREE,A ',I2,' . ')
          CALL ERTRAN(6,IMAGE)
          ENCODE(101,IFREE) IUSE
    101   FORMAT('@FREE   ',I2,' . ')
          ENCODE(102,ID) IUSE
    102   FORMAT(I2,10X)
          ISW = 0
    1     CONTINUE
          ENCODE(103,IMAGE) NAME
    103   FORMAT('@ASG,YZ ',A6,A4,'M. . ')
          CALL ERTRAN(6,IMAGE)
          ENCODE(104,IMAGE) IUSE, NAME
    104   FORMAT('@USE ',I2,',',A6,A4,'M. . ')
          CALL ERTRAN(6,IMAGE)
          CALL STATUS(ID,ID1,ID2,IFLAG)
          CALL DKSTAT(ID,ICODE1,ICODE2,JFLAG)
          CALL ERTRAN(6,IFREE)
          IF(IFLAG.EQ.0) GO TO 988
          IF(JFLAG.EQ.0) GO TO 988
          ENCODE(120,IMAGE)
    120   FORMAT('RESERVE 9-TRK DRIVE FOR CODAP TAPE FILE')
          CALL OMSG(7,IMAGE)
          ENCODE(122,IMAGE)
    122   FORMAT('THEN ENTER THE RESERVED TAPE DRIVE NUMBER (N)')
          LEN = 1
          CALL OMSGW(8,IMAGE,LEN,IUNIT)
          DO 3 ICHAR = 1,6
          IMAT = IFMT(ICHAR)
          DECODE(IMAT,IUNIT) JUNIT
          IF(JUNIT.EQ.IBLANK) GO TO 3
          CALL BCDBIN(JUNIT,JUNIT,1,$3)
          GO TO 4
    3     CONTINUE
          ENCODE(124,IMAGE)
    124   FORMAT('UNACCEPTABLE UNIT - PROGRAM ABORTS!')
          CALL OMSG(6,IMAGE)
          CALL ERTRAN(2)
    4     CONTINUE
          DO 5 INDEXA = 1,3
          ENCODE(105,IMAGE) NAME,JUNIT
    105   FORMAT('@ASG,UZ ',A6,A4,'M(+1).,S9/U',I1,' . ')
          WRITE(6,106) (IMAGE(I),I=1,5)
    106   FORMAT(1X,14A6)
          CALL DELAY(5)
          CALL GETFIL(IMAGE)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
              ENCODE(107,IMAGE) IUSE, NAME
       107    FORMAT('@USE ',I2,',',A6,A4,'M(+1). . ')
              WRITE(6,106) (IMAGE(I),I=1,5)
              CALL DELAY(5)
              CALL ERTRAN(6,IMAGE)
              IF(INDEXA.EQ.3) GO TO 5
              ENDFILE IUSE
              REWIND IUSE
              CALL DELAY(5)
              CALL FREE(IUSE)
              ENCODE(126,IMAGE) INDEXA,JUNIT
       126    FORMAT('DISMOUNT REEL  ',I1,'  OF  3  ON DRIVE ',I1)
              CALL OMSG(6,IMAGE)
       5      CONTINUE
              IONE = 1
              WRITE(2) IONE
              RETURN
       988    CONTINUE
              WRITE(6,108) NAME
       108    FORMAT(' FILE ',A6,A4,'M IS ALREADY CATALOGUED.')
              CALL BCDBIN(NAME(2),NUM,4,$989)
              NUM = NUM + 1
              ISW = ISW + 1
              IF(ISW.LT.6) GO TO 2
              WRITE(6,110)
       110    FORMAT(' RUN ABORTS AFTER 5 NAME CHANGE ATTEMPTS.')
              CALL ERTRAN(2)
       2      CALL BINBCD(NUM,NAME(2),4,$989,0)
              WRITE(6,111) NAME
       111    FORMAT(' CHANGING FILE NAME TO ',A6,A4,'M',/)
              GO TO 1
       989    CONTINUE
              WRITE(6,112) NAME
       112    FORMAT(' UNABLE TO ALTER FILE NAME ',A6,A4,'M.  RUN HALTS')
              CALL ERTRAN(2)
              ENTRY ASGCMR
              ENCODE(128,IMAGE) JUNIT
       128    FORMAT('PLEASE RELEASE DRIVE ',I1,' FROM RESERVE STATE')
              CALL OMSG(7,IMAGE)
              RETURN
              END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE AUTORV(NAME,IUSE)
      DIMENSION IMAGE(8), ID(2), NAME(2)
      ENCODE(101,ID) IUSE
101   FORMAT(I2,10X)
      ENCODE(102,IMAGE) NAME
102   FORMAT('@ASG,DAYQ ',A6,A4,'M(-0). . ')
      CALL ERTRAN(6,IMAGE)
      ENCODE(103,IMAGE) IUSE, NAME
103   FORMAT('@USE ',I2,',',A6,A4,'M(-0). . ')
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(ID,IREELS,ID2,IFLAG)
      IF(IFLAG.EQ.1) GO TO 999
10    CONTINUE
       ENCODE(106,IMAGE) NAME
106    FORMAT('@FREE   ',A6,A4,'M(-0). . ')
       CALL ERTRAN(6,IMAGE)
20     CONTINUE
      RETURN
999   CONTINUE
      WRITE(6,105) NAME
105   FORMAT(' AUTO RECOVERY FAILS. ',A6,A4,' IS MISSING FILE CYCLES.')
      CALL ERTRAN(2)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

BLANK BLANK OUT AN ARRAY (NWORD,IARRAY)

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE BLANK(NWORD,LIST)
      DIMENSION LIST(1)
      DO 1 IWORD = 1, NWORD
      LIST(IWORD) = ' '
1     CONTINUE
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...

$(0)      AXR$      .
ARG       FORM      12,6,18              . CONTROL WORD FORMAT
CKAD      RES       1                    . TEMPORARY STORE FOR AD
CKX11     RES       1                    . TEMPORARY STORE AREA FOR 'CKOPEN'
COMPKT1   C$OMPK,1  30,OPRMSG1,6,OPRESP .
COMPKT2   C$OMPK,1  48,OPRMSG2,6,OPRESP .
CSF       RES       10                   . DYNAMIC @ASG/@USE AREA
F                   '     ,F/D/XXX/      ' .
OPRMSG1             'WAITING FOR TAPE DRIVE.   AE' .
OPRMSG2             'DO YOU REALLY INTEND TO ERROR THIS RUN?   YES/NO' .
OPRESP    RES       1                    . OPERATOR'S RESPONSE TO QUERY
REEL                '   ,16N'            . USED FOR ASSIGNING MAG TAPE FILES
RN        RES       1                    . REEL NUMBER AND TEMPORARY STORAGE
SVA0      RES       1                    . CONTROL REGISTER SAVE AREA
SVA1      RES       1                    .
SVA2      RES       1                    .
SVA3      RES       1                    .
SVA4      RES       1                    .
SVX6      RES       1                    .
SVX7      RES       1                    .
SVX8      RES       1                    .
SVX10     RES       1                    .
SVX11     RES       1                    .
SVR1      RES       1                    .
$(1)      END$      .
/.        FILE-INFORMATION-TABLE (FIT) FOR CFHIO
 .
 .        01  FIT.
 .            02  FILE-CHARACTERISTICS.
 .                03  INT-FILE-NAME      PIC X(12).                 FIT+0
 .                03  EXT-FILE-NAME      PIC X(30).                 FIT+2
 .                03  BLOCK-SIZE         PIC H9(10).                FIT+7
 .                03  LRL                PIC H9(10).                FIT+8
 .                03  OPTIONS            PIC X(6).                  FIT+9
 .                03  ASSIGNMENT         PIC X(6).                  FIT+10
 .            02  FCT-AREA               PIC X(6) OCCURS 83.        FIT+11
 .            02  BUFFER-AREA            PIC X(6) OCCURS N.         FIT+94
 .
 .                                       N = (BLOCKSIZE + 2) * 2 FOR DOUBLE BUF.
 .                                       N = (BLOCKSIZE + 2) FOR SINGLE BUFFERING
 .
 .        NOTES:    IF EXT-FILE-NAME = SPACES, CFHIO WILL USE INT-FILE-NAME TO
 .                  ASSIGN THE FILE AND NO @USE WILL BE GENERATED.
 .
 .                  LRL IS ALWAYS SPECIFIED IN WORDS.  WHEN THE FILE IS CLOSED
 .                  VIA CLOSEF, THE LRL WILL BE SET TO ZERO.
 .
 .                  OPTIONS = S1.....INPUT/OUTPUT FLAG      (FIELDATA CHARACTERS)
 .                                   I = INPUT FILE
 .                                   O = OUTPUT FILE
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

```
LINE #   TEXT...

         .
         .
         .                        S2.....BUFFERING FLAG
         .                               D = DOUBLE BUFFERING
         .                               S = SINGLE BUFFERING
         .
         .                        S3.....LABELING FLAG
         .                               O = OMITTED LABELS
         .                               S = STANDARD LABELS
         .                               U = USER LABELS
         .
         .                        S4.....OPEN OPTIONS
         .                               N = NO REWIND
         .                               R = REWIND
         .
         .                        S5.....CLOSE OPTIONS
         .                               L = LOCK
         .                               N = NO REWIND
         .                               R = REWIND
         .
         .                        S6.....DYNAMIC ASSIGN FLAG
         .                               N = NO PRINT
        /.                               P = PRINT @ASG AND @USE STATEMENTS
         .
         .
         .            ASSIGNMENT = S1.....ASSIGNMENT FLAG
         .                                 C = CATALOGUE THE FILE
         .                                 T = TEMPORARY FILE
         .
         .                          S2.....SPACE REQUIREMENT
         .                                 F = FASTRAND (EXEC-8 DEFAULT AMOUNT)
         .                                 P = POS GRANULARITY
         .                                 T = TRK GRANULARITY
         .
         .                         S3-S6..NUMBER OF GRANULES (FD)
         .
        /.                         IF S1-S6 = SPACES, @ASG,A IS MADE.
         .     'OPENF' SUBROUTINE -- OPEN CFH FILE
         .
       OPENF.    .       CALL 'OPENF' USING FIT, WB.
                 .
                 .       LMJ       X11,OPENF
                 .       +         FIT                 0
                 .       +         WB                  1
                 .                 RETURN              2
                 .
                 .
                DS       A0,SVA0         . SAVE ALL REGISTERS USED BY OPENF
                DS       A2,SVA2         .
                SA       A4,SVA4         .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...

         SX       X6,SVX6          .
         SX       X7,SVX7          .
         SX       X8,SVX8          .
         SX       X10,SVX10        .
         SX       X11,SVX11        .
         SR       R1,SVR1          .
         LA       A0,0,X11         . A0 = FIT ADDRESS
         DL       A1,0,A0          .
         DS       A1,11,A0         . STORE INT-FILE-NAME IN FCT
         LR,U     R1,81            .
         LA       A2,(0,(0))       .
         AU,U     A0,13            . A1 = FIT+13 (FCT+2)
         LXI,U    A1,1             .
         BT       A1,0,*A2         . CLEAR REMAINDER OF FCT TO ZEROS
         LA,U     A1,01            .
         SA,S2    A1,21,A0         . SEQ = SEQUENTIAL FILE FLAG
         LA,S3    A2,9,A0          . A2 = LABELING FLAG
         TE,U     A2,+'U'          .
         LSSC     A1,6             . SHIFT FOR S OR O (A1 = 0100)
         TNE,U    A2,+'S'          .
         LSSC     A1,6             . STANDARD LABELS (A1 = 010000)
         SA,H2    A1,22,A0         . SLAB, OLAB OR ULAB = 01
         AU,U     A0,94            .
         SA,H2    A1,27,A0         . BUFF1 = FIT+94
         LA,S2    A2,9,A0          . A2 = BUFFERING FLAG
         TE,U     A2,+'D'          .
         J        $+4              . 'S' = SINGLE (DON'T SET BUFF2)
         AA       A1,7,A0          .
         AA,U     A1,2             .
         SA,H1    A1,27,A0         . BUFF2 = FIT+94+BLOCKSIZE+2
         DL       A1,7,A0          . A1 = BLOCKSIZE, A2 = LRL
         LSSL     A2,18            .
         DSL      A1,18            .
         SA       A2,30,A0         . SET (MAXBLK,MAXREC)
         TLE,U    A1,100           . JUMP IF USER'S OWN BSERI TO BE USED
         LA,U     A1,BSERI         . BEGINNING STANDARD EOR INPUT ADDRESS
         SA,H1    A1,57,A0         . SET BESRI ADDRESS IN FCT+056
         AA,U     A0,11            . A0 = FCT ADDRESS
         SA       A0,CKA0          . SAVE POINTER TO FCT
         LX       X10,0,X11        .
         LA       A4,2,X10         . A4 = EXT-FILE-NAME
         TNE      A4,('      ')    . IS IT BLANK?
         J        FACIL            . YES, DON'T BUILD @USE STATEMENT
         LMJ      X11,MOVFD        . NO SET CSF AREA FOR @USE
         ARG      4,0,('@USE  ')   .
         ARG      56,0,CSF         .
         LX       X7,X10           .
         LXI,U    X7,1             . X7 = (1,FIT)
         LX       X8,(1,CSF+2)     .
         LA       A2,(1,0)         .
```

LINE #   TEXT...

```
               LA        A3,(1,0)               .
               LR,U      R1,11                  .
               SLJ       CMOVE                  . MOVE INT-FILE-NAME
               LA,U      A4,+',' 	                .
               EX        PUTC,*A3               . STORE COMMA
               LXM       X7,X10                 .
               AX,U      X7,2                   . X7 = (1,FIT+2)
               LA        A2,(1,0)               .
               LR,U      R1,29                  .
               SLJ       CMOVE                  . MOVE EXT-FILE-NAME
               SLJ       CSFER                  . SEND @USE STATEMENT TO EXEC-8
      FACIL    LA        A0,CKA0                . RESET FCT ADDRESS
               ER        FACIL$                 . SEE IF FILE IS ALREADY ASSIGNED
               TZ,S1     17,X10                 .
               J         OPEN                   . FCT+6 IS NONZERO IF EQUIP IS ASSIGNED
               LMJ       X11,MOVFD              . INITIALIZE CSF AREA TO @ASG,A AND SPACES
               ARG       7,0,('@ASG,AZ          ') .
               ARG       53,0,CSF               .
               LX        X7,X10                 .
               LXI,U     X7,1                   . X7 = (1,FIT)                    INT-FILE-NAME
               LR,U      R1,11                  .
               LA        A1,2,X7                .
               TNE       A1,('      ')          .
               J         $+3                    .
               AX,U      X7,2                   . X7 = (1,FIT+2)                  EXT-FILE-NAME
               LR,U      R1,29                  .
               LX        X8,(1,CSF+2)           .
               LA        A2,(1,0)               .
               LA        A3,(1,0)               .
               SL        CMOVE                  . MOVE FILENAME TO @ASG STATEMENT
               LA        A1,10,X10              . A1 = ASSIGNMENT WORD
               TNE       A1,('      ')          . IS IT BLANK?
               J         DONE                   . YES, @ASG CARD IS COMPLETE
               LA,S1     A4,10,X10              . A4 = ASSIGNMENT
               SA,S6     A4,CSF                 . CLOBBER 'A' ON @ASG,A
               LR,U      R1,1                   . 'F' = CMOVE ',F' ONLY
               LA,S2     A4,10,X10              . A4 = GRANULARITY CODE (F,P,T,SPACE)
               LA,U      A2,+'TRK'              . T = TRK
               TNE,U     A4,+' '                .
               J         TAPE                   . BLANK = MAG TAPE ASSIGNMENT
               TNE,U     A4,+'P'                .
               LA,U      A2,+'POS'              . P = POS
               TE,U      A4,+'F'                .
               LR,U      R1,12                  .
               SA,H2     A2,F+1                 . STORE GRANULARITY
               LDSL      A0,42                  .
               AND       A0,(007777777700)      . KEEP ONLY NUMBER OF GRANULES (0NNNN0)
               OR        A1,(074000000005)      . MASK ON / AND SPACE (/NNNN )
               SA        A2,F+2                 .
               LX        X7,(1,F)               .
```

LINE #   TEXT...

```
                 LA        A2,(1,4)           .
                 J         DONE-1             .
        TAPE     LX        X7,(1,REEL)        .
                 LA        A2,(1,2)           .
                 LR,U      R1,3               .
                 SLJ       CMOVE              . MOVE REMAINDER OF @ASG CARD
        DONE     SLJ       CSFER              . SEND @ASG TO EXEC-8
        OPEN     LA,U      A0,0               . A0 = REWIND CODE
                 SZ        13,X10             . ZERO OUT THE IO PACKET INFO
                 SZ        14,X10             .
                 LA,S4     A4,9,X10           . A4 = OPEN OPTION
                 TNE,U     A4,+'N'            .
                 LA,U      A0,023             . A0 = NO-REWIND OPTION
                 LA,S1     A4,9,X10           . A4 = I/O FLAG
                 AX,U      X10,11             . X10 = FCT ADDRESS
                 TE,U      A4,+'I'            .
                 J         $+3                .
                 LMJ       X11,LOI$           . CFH OPEN INPUT ROUTINE
                 J         $+2                .
                 LMJ       X11,LOO$           . CFH OPEN OUTPUT ROUTINE
                 DL        A0,SVA0            . RESTORE ALL REGISTERS
                 DL        A2,SVA2            .
                 LA        A4,SVA4            .
                 LX        X6,SVX6            .
                 LX        X7,SVX7            .
                 LX        X8,SVX8            .
                 LX        X10,SVX10          .
                 LX        X11,SVX11          .
                 LR        R1,SVR1            .
                 J         2,X11              . RETURN
        /.       SPECIAL ROUTINES FOR 'OPENF'
                 .
                 .
        CMOVE    J         0                  . ENTRY AND EXIT
                 EX        GETC,*A2           . PICK UP ONE CHARACTER INTO A4
                 TNE,U     A4,+' '            . IS IT A SPACE?
                 J         CMOVE              . YES, EXIT IMMEDIATLY
                 EX        PUTC,*A3           . NO, STORE IT INTO CSF AREA
                 JGD       R1,$-4             . LOOP BACK FOR MORE
                 J         CMOVE              . RETURN
                 .
                 .
        CSFER    J         0                  . ENTRY AND EXIT
                 LA,U      A4,+' '            .
                 EX        PUTC,*A3           . STORE A SPACE
                 LA,U      A4,+'.'            .
                 EX        PUTC,*A3           . STORE PERIOD AT END FOR CSFS
                 LA,S6     A4,9,X10           . A4 = CSF PRINT FLAG
                 TE,U      A4,+'P'            . ARE WE SUPPOSED TO PRINT CSF STATEMENTS?
                 J         $+3                . NO, SKIP PRINTS
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
                LA      AO,(ARG 2,10,CSF) . YES, PRINT CSF AREA DOUBLE SPACED
                ER      PRINTS$           .
       CSFER1   LR,U    R1,7              . SET 2 MINUTE WAIT LOOP
                LA,U    AO,CSF            .
                ER      CSF$              . ISSUE DYNAMIC @ASG IMAGE
                JP      AO,CSFER          . EXIT IF NO FAC REJECT
                SA      AO,RN             . SAVE AO FOR ERROR MESSAGE
                TOP     AO,(1,0)          .
                J       CSFER2            . REJECT IS NOT FACILITIES WAIT
                LA,U    A1,15000          .
                ER      TWAIT$            . WAIT 15 SECONDS
                JGD     R1,CSFER1+1       . LOOP BACK 7 TIMES
                C$OM    COMPKT1           . DISPLAY "WAITING" MESSAGE
                LA,S1   AO,OPRESP         . AO = OPERATOR'S RESPONSE
                TE,U    AO,'E'            .
                J       CSFER1            . GO BACK INTO 2 MINUTE LOOP
                C$OM    COMPKT2           . DISPLAY "ERROR?" MESSAGE
                LA,S1   AO,OPRESP         . AO = OPERATOR'S RESPONSE
                TE,U    AO,'Y'            .
                J       CSFER1            . DON'T ERROR JOB AFTER ALL
       CSFER2   TNE,U   A4,+'P'           . HAVE WE ALREADY PRINTED THE BAD CARD?
                J       $+3               . YES, SKIP TO THE FAC REJECT CALL
                LA      AO,(ARG 2,10,CSF) . NO, PRINT IT NOW
                ER      PRINT$            .
                LMJ     X11,FACREJ        . PRINT FAC REJECT INFORMATION
                +       RN                . ERROR CODES
                +       0                 . DUMMY WALK-BACK
                J       CSFER             . EXIT AS IF NOTHING HAPPENED
                .
                .
       CKOPEN   J       0                 . ENTRY AND EXIT
                SX      X11,CKX11         . SAVE X11
                LX      X11,0,X11         . X11 = FIT ADDRESS
                TZ,H2   20,X11            . IS THIS FILE ALREADY OPEN?
                J       $+5               . YES, SKIP OVER THE CALL TO OPENF
                SX      X11,$+2           .
                LMJ     X11,OPENF         . OPEN THE FILE
                +       0                 . FIT ADDRESS FILLED IN ABOVE
                +       0                 .
                LX      X11,CKX11         . RESTORE X11
                J       CKOPEN            . RETURN
       /.       'READF' SUBROUTINE -- READ CFH FILE
        .
        .
       READF.   .       CALL 'READF' USING FIT, RECORD, EOF, WB.
                .
                .       LMJ       X11,READF
                .       +         FIT            0
                .       +         RECORD         1
                .       +         EOF            2
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
           *          *         WB                3
           *                    RETURN            4
           *
           *
           SLJ        CKOPEN               . SEE IF THIS FILE IS OPEN
           SA         A1,SVA1              . SAVE ALL REGISTERS USED BY READF
           SX         X7,SVX7              .
           SX         X8,SVX8              .
           SX         X10,SVX10            .
           SX         X11,SVX11            .
           SR         R1,SVR1              .
           LX         X7,1,X11             .
           LXI,U      X7,1                 . X7 = (1,RECORD)
           SX         X7,SVX6              . SAVE FOR LATER USE
           LX         X10,0,X11            . X10 = FIT ADDRESS
           LX,H1      X8,33,X10            . X8 = USESIZ
           AX,U       X10,11               . X10 = FCT ADDRESS
           LMJ        X11,LR$              . CFH READ ROUTINE
           +          EOF,$+1              .
           LX         X7,SVX6              . X7 = (1,RECORD)
           LR         R1,X8                . R1 = LENGTH OF PREVIOUS RECORD
           LX         X8,(0,('      ')) .
           BT         X7,0,*X8             . CLEAR PREVIOUS RECORD TO SPACES
           LX         X7,SVX6              . X7 = (1,RECORD)
           LXM        X8,22,X10            .
           LXI,U      X8,1                 . X8 = (1,USELOC)
           LR,H1      R1,22,X10            . R1 = USESIZ
           ANX,U      X10,11               . X10 = FIT ADDRESS
           SR         R1,8,X10             . STORE LENGTH OF RECORD IN FIT+8
           BT         X7,0,*X8             . MOVE RECORD FROM BUFFER TO USER AREA
           SZ         SVX6                 .
           J          $+3                  .
   EOF     ANX,U      X10,11               . X10 = FIT ADDRESS
           SZ         8,X10                . SET LRL = 0
           LA         A1,SVA1              . RESTORE ALL REGISTERS
           LX         X7,SVX7              .
           LX         X8,SVX8              .
           LX         X10,SVX10            .
           LX         X11,SVX11            .
           LR         R1,SVR1              .
           TNZ        SVX6                 . WAS THIS AN EOF CONDITION?
           J          4,X11                . NO, TAKE NORMAL RETURN
           TZ         *2,X11               . IS THERE AN ALTERNATE EXIT?
           J          *2,X11               . YES, TAKE THE USER'S EOF EXIT
           J          4,X11                . NO, TAKE NORMAL RETURN POINT
   /.      BEGINNING OF STANDARD END OF REEL INPUT ROUTINE
   .
   .
   BSERI   J          0                    . ENTRY AND EXIT
           LX         X11,SVX11            .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
               LX        X11,0,X11            . X11 = FIT ADDRESS
               LA,H1     A1,7,X11             . A1 = USER'S REEL COUNT
               JZ        A1,BSER1             . IMMEDIATE EXIT IF IT IS ZERO
               ANA,U     A1,1                 .
               SA,H1     A1,7,X11             . NREEL = NREEL - 1
               JZ        A1,EOF               . EXIT ON ZERO REEL COUNT
               J         BSER1                . NORMAL RETURN FOR REEL SWITCHING
     /.        'WRITEF' SUBROUTINE -- WRITE CFH FILE
      .
      .
     WRITEF*   .         CALL 'WRITEF' USING FIT, RECORD, WB.
               .
               .         LMJ       X11,WRITEF
               .         +         FIT                 0
               .         +         RECORD              1
               .         +         WB                  2
               .                   RETURN              3
               .
               .
               SLJ       CKOPEN               . SEE IF THIS FILE IS OPEN
               SA        A0,SVA0              . SAVE ALL REGISTERS USED BY WRITEF
               SX        X7,SVX7              .
               SX        X8,SVX8              .
               SX        X10,SVX10            .
               SX        X11,SVX11            .
               SR        R1,SVR1              .
               LX        X10,0,X11            . X10 = FIT ADDRESS
               LR        R1,8,X10             . R1 = LRL (WORDS)
               SR        R1,SVX6              . SAVE FOR LATER USE
               LX        X8,1,X11             .
               LXI,U     X8,1                 . X8 = (1,RECORD)
               AX,U      X10,11               . X10 = FCT ADDRESS
               LXM       X7,22,X10            .
               LXI,U     X7,1                 . X7 = (1,USELOC)
               BT        X7,0,*X8             . MOVE RECORD INTO BUFFER
               LA        A0,SVX6              . A0 = LRL (WORDS)
               MSI,U     A0,6                 . A0 = LRL (CHARACTERS)
               LMJ       X11,LW$              . CFH WRITE ROUTINE
               +         $+1                  .
               LA        A0,SVA0              . RESTORE ALL REGISTERS
               LX        X7,SVX7              .
               LX        X8,SVX8              .
               LX        X10,SVX10            .
               LX        X11,SVX11            .
               LR        R1,SVR1              .
               J         3,X11                .
     /.        'CLOSEF' SUBROUTINE -- CLOSE CFH FILE
      .
      .
     CLOSEF*   .         CALL 'CLOSEF' USING FIT, WB.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

CFHIO COBOL FILE HANDLER SUBROUTINE

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
          .      LMJ      X11,CLOSEF
          .      +        FIT                0
          .      +        WB                 1
          .               RETURN             2
          .
          .
          .
          DS     A0,SVA0            . SAVE ALL REGISTERS USED BY CLOSEF
          SX     X10,SVX10          .
          SX     X11,SVX11          .
          LX     X10,0,X11          . X10 = FIT ADDRESS
          SZ     8,X10              . SET LRL = 0
          TNZ,H2 20,X10             . IS THIS FILE OPEN?
          J      NOCLOSE            . NO, SKIP CLOSE ROUTINE
          AX,U   X10,11             . X10 = FCT ADDRESS
          LA,S5  A1,9,X10           . A1 = CLOSE OPTION
          LA,U   A0,0               . A0 = 'REWIND'
          TNE,U  A1,+'L'            .
          LA,U   A0,021             . A0 = 'LOCK'
          TNE,U  A1,+'N'            .
          LA,U   A0,023             . A0 = 'NO REWIND'
          LMJ    X11,LCFS           . CFM CLOSE ROUTINE
NOCLOSE   DL     A0,SVA0            . RESTORE ALL REGISTERS
          LX     X10,SVX10          .
          LX     X11,SVX11          .
          J      2,X11              . RETURN
/.        LITERALS
.
.
          END    .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE CPI(NR,IAXP,LAP,LFW,NFC,KTFW,KTFC,ICMA,LOPT)
      DIMENSION MAT(3)
      COMMON/FGEN/ IFMT(1000), DUMMY(4504)
C  THIS SUBROUTINE PACKS FORMAT CHARACTERS AND NUMBERS LEFT ADJUSTED
C  INTO A SINGLE COMPUTER WORD.
C    EXAMPLE - INPUT • ICMA=, NR=3 IAXP=A LAP=2
C                OUTPUT• LFW=,3A2   NFC=4
      DATA LETA, LETI, LETX/'A','I','X'/,ICOMA/','/
      IF(IAXP.EQ.'F') GO TO 7
      IF(IAXP.EQ.LETI.AND.(LOPT.EQ.0.OR.LOPT.EQ.2)) IAXP = LETA
      IF(IAXP.EQ.LETX.AND.LOPT.EQ.2) IAXP = LETA
C  ESTABLISH ENCODE FORMAT TYPE
      NI = 2
      IF(IAXP.EQ.LETX) NI = 1
C  DETERMINE THE NUMBER OF DIGITS IN NR
      CALL BINBCD(NR,NUM,1,$3,1)
      NFC = NI + 2
      GO TO (1,2), NI
 1    ENCODE(101,LFW) ICMA, NUM, IAXP
      GO TO 6
 2    ENCODE(101,LFW) ICMA, NUM, IAXP, LAP
      GO TO 6
 3    CALL BINBCD(NR,NUM,2,$99,1)
      NFC = NI + 3
      GO TO (4,5), NI
 4    ENCODE(102,LFW) ICMA, NUM, IAXP*
      GO TO 6
 5    ENCODE(102,LFW) ICMA, NUM, IAXP, LAP
 6    CALL INSERT(KTFW,KTFC,LFW,NFC)
      ICMA = ICOMA
      RETURN
 7    CONTINUE
      CALL BINBCD(NR,NUM,1,$8,1)
      NL1 = 1
      GO TO 9
 8    CALL BINBCD(NR,NUM,2,$99,1)
      NL1 = 2
 9    CALL BINBCD(LAP,LR,1,$10,1)
      NL2 = 1
      GO TO 11
 10   CALL BINBCD(LAP,LR,2,$99,1)
      NL2 = 2
 11   CONTINUE
      NFC = NL1 + NL2 + 2
      ENCODE(104,MAT) NL1,NL2
 104  FORMAT('(A1,A',I1,',A1,A',I1,')')
      ENCODE(MAT,LFW) ICMA,NUM,IAXP,LR
      CALL INSERT(KTFW,KTFC,LFW,NFC)
      ENCODE(105,LFW)
 105  FORMAT('.0')
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

CPI PACKS FORMAT CHARACTERS INTO A SINGLE WORD

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
        NFC = 2
        CALL INSERT(KTFW,KTFC,LFW,NFC)
        ICMA = ICOMA
        RETURN
99      WRITE(6,103)
        STOP
103     FORMAT('1BINARY VALUE TOO LARGE TO ENCODE INTO TWO CHARACTERS')
101     FORMAT(3A1,I1)
102     FORMAT(A1,A2,A1,I1)
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE CYCLES(NAME,IUSE,IASGD)
      DIMENSION NAME(2), ID(3), IMAGE(14), IFREE(3), IUSE2(2), NAM(3)
      DATA IBLANK /' '/
      IF(NAME(1).EQ.IBLANK) GO TO 999
      IF(NAME(1).EQ.'SCRATC') GO TO 999
      ENCODE(102,IFREE) IUSE
102   FORMAT('@FREE  ',I2,' . ')
      ENCODE(112,IMAGE) IUSE
112   FORMAT('@FREE,A ',I2,' . ')
      CALL ERTRAN(6,IMAGE)
      ENCODE(122,IUSE2) IUSE
122   FORMAT(I2,10X)
      ENCODE(140,ID) NAME
140   FORMAT(A6,A4,'M(-3)')
      ENCODE(106,IMAGE) ID
      CALL ERTRAN(6,IMAGE)
      ENCODE(103,IMAGE) IUSE, ID
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(IUSE2,ID1,ID2,IFLAG)
      CALL ERTRAN(6,IFREE)
      NAM(1) = NAME(1)
      NAM(2) = NAME(2)
      NAM(3) = NAME(3)
      IF(NAM(3).EQ.IBLANK) NAM(3) = '0'
      IF(IFLAG.EQ.1) GO TO 145
      WRITE(6,141) NAME
141   FORMAT(' AUTO RECOVERY ON FILE ',A6,A4)
      CALL AUTORV(NAM,IUSE)
145   ENCODE(105,ID) NAME
      ENCODE(103,IMAGE) IUSE, ID
103   FORMAT('@USE  ',I2,',',2A6,A3,' . ')
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(IUSE2,ID1,ID2,IFLAG)
      IASGD = 1
      IF(IFLAG.EQ.0) GO TO 900
      IASGD = 0
      CALL ERTRAN(6,IFREE)
      ENCODE(105,ID) NAME
105   FORMAT(A6,A4,'M(-2)')
      ENCODE(106,IMAGE) ID
106   FORMAT('@ASG,QY ',2A6,A3,' . ')
      CALL ERTRAN(6,IMAGE)
      ENCODE(103,IMAGE) IUSE, ID
      CALL ERTRAN(6,IMAGE)
      CALL STATUS(IUSE2,ID1,ID2,IFLAG)
      CALL ERTRAN(6,IFREE)
      IF(IFLAG.NE.0) GO TO 500
      IF(ID2.NE.0) GO TO 750
      IOLD = 28
         IF(IUSE.EQ.25) IOLD = 27
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      CALL ASGAM(NAM,IOLD)
132   FORMAT('@ASG,AD ',2A6,A3,' . ')
133   FORMAT('@CAT,OV ',A6,A4,'M(+1),16N,',A6,' . ')
      ENCODE(132,IMAGE) ID
      WRITE(6,119) (IMAGE(I),I=1,5)
      CALL GETFIL(IMAGE)
      ENCODE(131,IMAGE) IUSE, ID
131   FORMAT('@USE ',I2,',',2A6,A3,'. . ')
      WRITE(6,119) (IMAGE(I),I=1,5)
      CALL ERTRAN(6,IMAGE)
      ENCODE(133,IMAGE) NAME, ID1
      WRITE(6,119) (IMAGE(I),I=1,6)
      CALL ERTRAN(6,IMAGE)
      RETURN
500   CONTINUE
119   FORMAT(1X,8A6)
      WRITE(6,125) NAME
125   FORMAT(1X,A6,A4,' DOES NOT HAVE A (-2) CYCLE.  RUN HALTS.')
      CALL ERTRAN(2)
750   CONTINUE
      WRITE(6,108) NAME
108   FORMAT(' FILE ',A6,A4,' EXCEEDS ONE REEL.  RERUN TSETUP')
      CALL ERTRAN(2)
999   CONTINUE
      WRITE(6,110)
110   FORMAT(' INVALID FILENAME WAS REQUESTED FOR THIS FILE')
      CALL ERTRAN(2)
900   CONTINUE
      WRITE(6,166) ID, IUSE
166   FORMAT(' FILE ',2A6,A3,' ALREADY ASSIGNED. USE = UNIT ',I2)
      RETURN
      END
```

433a

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
              AXR$
$(0)
BLANK                    ' '
FDZERO                   '000000'
SV            RES        41
/. PROCEDURE
P             PROC
SAVE.         NAME
K             DO P(1) , 5     P(1,K),SV+K-1
              GO         OUT
REST.         NAME
L             DO P(1) , L     P(1,L),SV+L-1
OUT           NAME
              END
/. BCD TO BINARY ROUTINE
$(1)
BCDBIN.       SAVE       A11,A12,A13,A14,A15
              SZ         A12
              LA         A11,*2,X11
              TLE,U      A11,7
              TLE,U      A11,1
              J          BDBN4
              ANA,U      A11,1
              LA         A15,*0,X11
LOOP          SZ         A14
              LDSL       A14,6
              TE,U       A14,5
              J          BDBN2
              JGD        A11,LOOP
              J          BDBN3+1
BDBN1         SZ         A14
              LDSL       A14,6
BDBN2         TLE,U      A14,072
              TLE,U      A14,060
              J          BDBN4
              LDSL       A13,36
              AND,U      A13,017
              MSI,U      A12,10
              AA         A12,A14
BDBN3         JGD        A11,BDBN1
              SA         A12,*1,X11
              REST       A11,A12,A13,A14,A15
              J          5,X11
BDBN4         REST       A11,A12,A13,A14,A15
              J          *3,X11
/. BINARY TO BCD ROUTINE
BINBCD.       SAVE       A12,A13,A14,A15
              DSL        A14,72
              LA         A12,*0,X11
              TP         A12
```

433B

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
                J        BNBD3              .
        BNBD1   DSL      A12,36             .
                DI,U     A12,10             .
                DSL      A13,6              .
                AA,U     A15,1              .
                TZ       A12                .
                J        BNBD1              .
                LA       A12,*2,X11         .
                TLE,U    A12,7              .
                TLE,U    A12,1              .
                J        BNBD3              .
                ANA      A12,A15            .
                OR       A14,FDZERO         .
                TNZ      A12                .
                J        BNBD2              .
                TP       A12                .
                J        BNBD3              .
                MSI,U    A12,6              .
                SA,H2    A12,BNBD2-1  .
                LA       A14,FDZERO         .
                TZ       *4,X11             .
                LA       A14,BLANK          .
                DSL      A14,0    .
        BNBD2   SA       A15,*1,X11         .
                REST     A12,A13,A14,A15    .
                J        6,X11              .
        BNBD3   REST     A12,A13,A14,A15    .
                J        *3,X11             .
                END                         .
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      SUBROUTINE DATETM(LDATE,LTIME)
      DIMENSION LDATE(2), LTIME(2)
      DIMENSION  MON(12)
      DATA MON/'JAN','FEB','MAR','APR','MAY','JUN','JUL',
     1 'AUG','SEP','OCT','NOV','DEC'/
      CALL ERTRAN(9,IDATE,ITIME)
      DECODE(201,IDATE) MONTH
201   FORMAT(I2)
      IYEAR = FLD(12,24,IDATE)
      JDATE = FLD(12,12,IDATE)
      IMINS = FLD(12,12,ITIME)
      ENCODE(202,LTIME) ITIME, IMINS, ITIME
202   FORMAT(A2,':',R2,':',R2,4X)
      ENCODE(203,LDATE) JDATE,MON(MONTH),IYEAR
203   FORMAT(R2,1X,A3,1X,R2,3X)
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

DELAY   CAUSE PROGRAM TO WAIT SPECIFIED # OF SECONDS

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
$(0)    AXR$
SV      RES      1              .   ESTABLISH WORD TO SAVE REGISTER
$(1)    ERU$
DELAY*  SA       A1,SV          .   SAVE REGISTER A1
        LA       A1,*0,X11      .   RETREIVE TIME DELAY IN SECONDS
        MSI,U    A1,1000        .   CONVERT TO MILLISECONDS
        TG,U     A1,30000       .   SEE IF REQUEST EXCEEDS MAX
        LA,U     A1,30000       .       IF SO, SET REQUEST TO MAX ALLOWED
        ER       TWAIT$         .   EXECUTE THE WAIT
        LA       A1,SV          .   RESTORE REGISTER A1
        J        2,X11          .   RETURN TO USER
        END
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE DISK(NREC,LENGTH)
      DIMENSION LOC(1), IMAGE(2)
      DATA IMAGE/'@FREE 29 . '/
      CALL ERTRAN(6,IMAGE)
      IO = 29
      DEFINE FILE IO(NREC,LENGTH,U,NEXT)
      RETURN
      ENTRY DSKSTO(NWORD,LOC,IREC,ISW)
      JSW = 1
      IF(ISW.EQ.0) GO TO 1
      WRITE(IO'IREC,ERR=100) (LOC(I),I=1,NWORD)
      RETURN
      ENTRY DSKRED(NWORD,LOC,IREC,ISW)
      JSW = 2
      IF(ISW.EQ.0) GO TO 1
      READ(IO'IREC,ERR=100) (LOC(I),I=1,NWORD)
      RETURN
1     K = 0
      JREC = IREC - 1
      LEFT = MOD(NWORD,465)
      LOOP = NWORD/465
      IF(LOOP.EQ.0) GO TO 3
      DO 2 J = 1, LOOP
      JREC = JREC + 1
      L = K + 1
      K = K + 465
      IF(JSW.EQ.1) WRITE(IO'JREC,ERR=100) (LOC(I),I=L,K)
      IF(JSW.EQ.2)  READ(IO'JREC,ERR=100) (LOC(I),I=L,K)
2     CONTINUE
3     IF(LEFT.EQ.0) RETURN
      K = K + 1
      JREC = JREC + 1
      IF(JSW.EQ.1) WRITE(IO'JREC,ERR=100) (LOC(I),I=K,NWORD)
      IF(JSW.EQ.2)  READ(IO'JREC,ERR=100) (LOC(I),I=K,NWORD)
      RETURN
100   WRITE(6,101)
101   FORMAT('1ATTEMPTED TO RANDOMLY READ/WRITE RECORD BEYOND ALLOCATED
     1DISK LIMITS')
      CALL ERTRAN(2)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

DKSTAT SEE IF A MASS STORAGE FILE IS ASSIGNED

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
$(0)    AXR$
PKT     RES     13
PF      FORM    12,6,18
SV      RES     3
$(1)    ERU$
DKSTAT*
        DS      A0,SV       .  SAVE A0,A1
        SZ      PKT+6       .  SET EQUIPMENT & MODE CODE TO ZERO
        DL      A0,*0,X11     .  RETREIVE FILENAME
        DS      A0,PKT        .  STORE FILENAME IN PACKET
        LA      A0,(10,PKT)
        ER      FITEM$
        LA,S1   A0,PKT+6    .     PICK UP EQUIPMENT CODE
        TNZ,S1  PKT+6           .   TEST TO SEE IF EQUIPMENT CODE IS ZERO
        J       ERROR           .   IF SO, JUMP TO ERROR ROUTINE
        TNE,U   A0,014          .   SEE IF EQUIPMENT IS 16N TAPE
        J       ERROR           .   IF SO, JUMP TO ERROR ROUTINE
        SA      A0,*1,X11       .   RETURN EQUIPMENT CODE
        LA,S2   A0,PKT+6         .    PICK UP FILE MODE
        SA      A0,*2,X11       .   RETURN FILE MODE
        LA,S1   A0,PKT+7         .    PICK UP TEMP/CAT FLAG
        SSL     A0,4          .      SHIFT 'DOWN' BIT OUT OF WORD
        LSSL    A0,35         .      GET RID OF LABEL CHECK BIT
        SSL     A0,35       .      RETURN TEMP/CAT FLAG
        SA      A0,*3,X11   .      GIVE THE FLAG TO THE MAIN PROGRAM
        DL      A0,SV       .  RESTORE REGISTERS
        J       5,X11       .    RETURN
ERROR   SZ      *1,X11          .   SET EQUIPMENT CODE TO ZERO
        SZ      *2,X11          .   SET FILE MODE TO ZERO
        LA,U    A0,1            .   SET A0 TO ONE
        SA      A0,*3,X11       .   SET TEMP/CAT FLAG TO 1 (NOT ASSIGNED)
        DL      A0,SV           .   RESTORE REGISTERS
        J       5,X11           .   RETURN TO USER
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
          AXR$
          ERU$ .
 $(0)
 P           PROC
 SAVE*       NAME                                .
 K           DO P(1) , S     P(1,K),SV+K-1       .
             GO          OUT                     .
 REST*       NAME                                .
 L           DO P(1) , L     P(1,L),SV+L-1       .
 OUT         NAME                                .
             END                                 .
 PKT     I$OD      '28       ',R$ 224,BUF 0000
 BUF     RES       252
 WRDCNT  RES       1
 BLOCK   RES       1
 SV      RES       5
 MESAGE            '000000 IS INVALID. BLOCK/WORD                DECODE'
 ERROR             '        = STATUS. SEE UP4144 PAGE C-17'
 UNEXPT            'UNEXPECTED END OF FILE ON REPORT FILE'
 SX11      +          0
 NEED    +         0
 TEST              'BEGIN REPORT'
 BLONUM    +          0
 BLANK             '   '
 WB                'REWIND'
           +       0
 PF      FORM      12,6,18
 $(1)
 DREAD*
         SAVE      A0,A2,A5,X2,X3 . SAVE CONTENTS OF CONTROL REGISTERS
         L         A5,NEED        . RETREIVE NEED NEW BLOCK FLAG
         TZ        A5             . IF NEEDED, CONTINUE
         J         DECODE         .     ELSE JUMP TO DECODE SDF CONTROL WORD
 RENTER  SZ        WRDCNT
 OK1     LMJ       X2,READ        . JUMP TO READ A NEW BLOCK
         TZ        A2             . TEST STATUS WORD
         J         ERR            .    IF NON-ZERO JUMP TO FIND CAUSE
         LX,U      X3,1           . SET X3 = 1
         SX        X3,NEED        . CANCEL NEED FOR NEW BLOCK
 DECODE  LX        X3,WRDCNT      . RETREIVE NUMBER OF WORDS PROCESSED
 .                  THIS WILL ALSO SET POINTER TO SDF CONTROL WORD
         L,S1      A2,BUF,X3      . RETREIVE SDF CODE FROM CONTROL WORD
         TNZ       A2             . IF NOT A DATA IMAGE, CONTINUE
         J         DATA           .     ELSE JUMP TO PROCESS DATA
         TNE,U     A2,050         . IF NOT A LABEL IMAGE, CONTINUE
         J         LABEL          .     ELSE JUMP TO BYPASS LABEL
         TNE,U     A2,077         . IF NOT END-OF-FILE, CONTINUE
         J         EOFXIT         .     ELSE TAKE EOF EXIT
         TNE,U     A2,040         . IF NOT PAD RECORD, CONTINUE
         J         RENTER         .     ELSE JUMP TO RE-ENTER SUBROUTINE
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
         LA       A2,BUF,X3        . RETREIVE ENTIRE CONTROL WORD FOR MESSAGE
         S        A2,MESAGE        . PREPARE ERROR MESSAGE
           SX        X11,SX11           . SAVE X11
           LMJ       X11,MOVBIN         . CALL DICK'S MOVE/CONVERT ROUTINE
           PF        1,0,MESAGE+5       . FORMS CONTROL W/ARGS
           PF        0,0,BLONUM         . FORMS CONTROL W/ARGS
           LMJ       X11,MOVBIN         . CALL DICK'S MOVE/CONVERT ROUTINE
           PF        2,0,MESAGE+6       . FORMS CONTROL W/ARGS
           PF        0,0,WRDCNT         . FORMS CONTROL W/ARGS
           LX        X11,SX11           . RESTORE X11
           LA        A0,(PF 1,8,MESAGE) . LOAD ERROR MESAGE
           ER        PRINT$             . PRINT THE MESSAGE
           LA        A2,BLONUM          . GET BLOCK NUMBER
           TNE,U     A2,0               . IF BLOCK NUMBER NOT = 0, CONTINUE
           J         RENTER             .    ELSE JUMP TO RE-ENTER
           TNE,U     A2,1               . IF BLOCK NE 1, CONTINUE
           J         OK1                .    ELSE JUMP TO OK1
           TNE,U     A2,2                . IF BLOCK NUMBER NOT = 2, CONTINUE
           J         OK1                .    ELSE JUMP TO OK1
         LMJ      X2,READ          . JUMP TO READ A NEW BLOCK
         TNE,U    A2,1             . IF NOT END OF FILE, ABORT WITH MESSAGE
         J        EOFXIT           .     ELSE TAKE END OF FILE RETURN
         ER       ERR$             . STOP RUN
LABEL    AX,S2    X3,BUF,X3         . ADD LABEL LENGTH TO WORDS PROCESSED
         AX,U     X3,1             . INCREMENT FOR LABEL CONTROL WORD
         SX       X3,WRDCNT        . STORE NEW WORD COUNT
         J        DECODE           . SKIP THIS LABEL IMAGE
ERR      TNE,U    A2,1             . IF NOT END OF FILE, CONTINUE
         J        EOFXIT           .     ELSE RETURN THRU EOF EXIT
ERR1     SX       X11,SX11         . SAVE X11
           LMJ       X11,BINOCT         . MOVE & CONVERT TO FD OCTAL
           PF        1,0,ERROR          . MOVE TO ERROR MES
           PF        0,0,A2             . PICK UP ERROR CODE
           LX        X11,SX11           . RESTORE X11
         L        A0,(PF 1,7,ERROR) . LOAD ERROR MESSAGE
         ER       PRINT$           . PRINT IT
         ER       ERR$             . STOP
EOFXIT   REST     A0,A2,A5,X2,X3          . RESTORE REGISTERS FOR RETURN
         J        4,X11                   . RETURN THRU EOF ADDRESS
DATA     L,S2     A2,BUF,X3        . RETREIVE LENGTH OF DATA IMAGE
         TNZ      A2               . IF NOT ZERO, CONTINUE
         J        LENGTH           .     ELSE JUMP TO LENGTH ERROR MESSAGE
         S        A2,*0,X11        . TRANSMIT LENGTH TO CALLING PROGRAM
         AX,U     X3,1             . ADJUST POINTER TO LOCATE FIRST IMAGE WORD
         A        A2,X3            . ADD IMAGE LENGTH TO WORD COUNT
         TG       A2,BLOCK         . IF BLOCKSIZE EXCEEDS WORD COUNT, CONTINUE
         J        SPAN             .     ELSE JUMP TO SPANNED RECORDS
         S        A2,WRDCNT        . SAVE NEW WORD COUNT
SEND     LR       R1,*0,X11        . RETREIVE LENGTH INTO REPEAT COUNTER
         LX       X2,1,X11         . STORE ADDRESS OF RECEIVING ARRAY
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...

 101           LXI,U   X2,1               . ESTABLISH INCREMENTATION OF SUBSCRIPTS
 102           AX      X3,(1,BUF)         . ESTABLISH SENDING ADDRESSES
 103           BT      X2,,*X3            . TRANSMIT THE IMAGE
 104   CHECK   L       A2,*0,X11          . RETREIVE IMAGE LENGTH
 105           TE,U    A2,3               . IF EQUAL TO 3, CONTINUE
 106           J       NORMAL             .     ELSE TAKE NORMAL RETURN
 107           L       A2,*1,X11          . RETREIVE FIRST WORD OFIMAGE
 108           TNE     A2,TEST            . IF IT IS NOT = 'BEGIN ', CONTINUE
 109           J       TEST2              .     ELSE JUMP TO ANOTHER TEST
 110   NORMAL  REST    A0,A2,A5,X2,X3             . RESTORE REGISTERS FOR RETURN
 111           J       6,X11                      . RETURN TO CALLER
 112   TEST2   LX      X3,1,X11           . RETREIVE WORD 2 OF IMAGE
 113           L       A2,1,X3            .    AND PLACE IN A2
 114           TE      A2,TEST+1          . IF IT IS EQUAL TO 'REPORT', CONTINUE
 115           J       NORMAL             .     ELSE JUMP TO NORMAL RETURN
 116           L       A2,2,X3            . RETREIVE 3RD WORD = J SEQUENCE NUMBER
 117           S       A2,*2,X11          . STORE SEQUENCE NUMBER
 118           REST    A0,A2,A5,X2,X3             . RESTORE REGISTERS FOR RETURN
 119           J       3,X11              . TAKE NEW REPORT RETURN
 120   LENGTH LA,S4    A2,BUF,X3 .
 121           TNE,U   A2,010 .
 122           J       BLINE
 123           TNE,U   A2,030
 124           J       EOFXIT
 125           TNE,U   A2,040                . IF FORTRAN CODE NOT A PAD IMAGE, CONTINUE
 126                                         .  (SEE COMMENTS CIRCA LINE 150)
 127           J       READN1                .    JUMP TO READNX TO HANDLE PAD IMAGE (&FIX X3)
 128           LA      A2,BUF,X3          . RETREIVE SDF CONTROL WORD
 129           SA      A2,MESAGE          . SETUP ERROR MESSAGE
 130              SX        X11,SX11              . SAVE X11
 131              LMJ       X11,MOVBIN            . CALL DICK'S MOVE/CONVERT ROUTINE
 132              PF        1,0,MESAGE+5          . FORMS CONTROL W/ARGS
 133              PF        0,0,BLONUM            . FORMS CONTROL W/ARGS
 134              LMJ       X11,MOVBIN            . CALL DICK'S MOVE/CONVERT ROUTINE
 135              PF        2,0,MESAGE+6          . FORMS CONTROL W/ARGS
 136              PF        0,0,WRDCNT            . FORMS CONTROL W/ARGS
 137              LX        X11,SX11              . RESTORE X11
 138              LA        A2,('LENGTH')
 139              SA        A2,MESAGE+7           . IDENTIFY PLACE OF ERROR
 140           LA      A0,(PF 1,8,MESAGE)   .  ESTABLISH PACKET ADDRESS
 141           ER      PRINT$          . PRINT THE MESSAGE
 142             TG,U      A2,3                . IF C(A2) IS LESS OR EQUAL 2 RENTER
 143           ER      ERR$            . STOP
 144             J         RENTER
 145   BLINE   LA      A2,BLANK        . SET A2 EQUAL TO WORD OF BLANKS
 146           SA      A2,*1,X11       . STORE BLANKS INTO LINE IMAGE
 147           LA,U    A2,1            . SET A2 = 1
 148           SA      A2,*0,X11       . SET IMAGE LENGTH = 1
 149           AX,U    X3,1            . INCREMENT POINTER TO CONTROL WORD
 150           SX      X3,A2           . SET A2 = POINTER
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

DREAD   READ A FORMATTED-FORTRAN WRITTEN MASS STORAGE FILE

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
         TG      A2,BLOCK        . TEST POINT GREATER THAN BLOCKSIZE
         SZ      NEED            . IF SO, SET NEED NEW BLOCK FLAG
         SX      X3,WRDCNT .
         J       NORMAL
SPAN     TNE     A2,BLOCK        . IF WORD COUNT IS NOT = BLOCKSIZE CONTINUE
         J       READNX          .     ELSE JUMP TO SET NEED TO READ BLOCK
         ANA     A2,BLOCK        . WORD COUNT - BLOCKSIZE = SPAN LENGTH
         SA      A2,WRDCNT       . SPAN LENGTH = WORD COUNT OF NEXT BLOCK
         L       A5,*0,X11       . RETREIVE LENGTH
         ANA     A5,A2           . LENGTH - SPAN = FIRST PORTION LENGTH
         LR      R1,A5           . LOAD REPEAT COUNTER WITH THAT LENGTH
         LX      X2,1,X11        . LOAD ARRAY ADDRESS
         LXI,U   X2,1            . LOAD SUBSCRIPT INCREMENT
         AX      X3,(1,BUF)      . LOAD BUFFER ADDRESS
         BT      X2,,*X3         . TRANSMIT IMAGE
         LMJ     X2,READ         . JUMP TO READ NEXT BLOCK
         TZ      A2              . IF SUCCESSFUL READ, CONTINUE
         J       BAD             .     ELSE JUMP TO BAD BLOCK
         LR      R1,WRDCNT       . LOAD SPAN LENGTH INTO REPEAT COUNTER
         LX      X2,1,X11        . LOAD ARRAY ADDRESS
         AX      X2,A5           . BIAS ADDRESS FOR SECOND PORTION TRANSFER
         LXI,U   X2,1            . LOAD SUBSCRIPT INCREMENT
         LX      X3,(1,BUF)      . LOAD BUFFER ADDRESS
         BT      X2,,*X3         . TRANSMIT IMAGE
         J       CHECK           . JUMP TO CHECK FOR SENTINEL
BAD      TE,U    A2,1            . IF END OF FILE READ, CONTINUE
         J       ERR1            .     ELSE JUMP TO TAPE STATUS ERROR
         L       A0,(PF 1,7,UNEXPT)  . LOAD ERROR MESSAGE
         ER      PRINT$          . PRINT IT
         ER      ERR$            . STOP
READN1   AX,U    X3,1            . FUDGE X3 FOR NEXT OPERATION
READNX   SZ      NEED            . SET NEED NEW BLOCK FLAG
         LA      A2,X3           . RETREIVE POINTER
         ANA,U   A2,1            . RESTORE POINTER TO CONTROL WORD
         LA,S4   A5,BUF,A2       . ACCESS FORTRAN CONTROL BYTE
                                 .     010 = "C" = NORMAL DATA IMAGE
                                 .     030 = "S" = END-OF-FILE
                                 .     040 = "(" = PAD IMAGE TO FILL OUT TAPE BLOCK
                                 . IF CODE NOT A PAD IMAGE, CONTINUE
         TNE,U   A5,040          .     ELSE JUMP TO RE-ENTER & GRAB A BLOCK
         J       RENTER          . JUMP TO SEND IMAGE TO CALLER
         J       SEND            . SETUP THE I/O PACKET
READ     L,U     A0,PKT          . ISSUE THE READ AND WAIT COMMAND
         ER      IOW$
            LA       A2,BLONUM          . GET BLOCK NUMBER
            AA,U     A2,1               . INCREMENT BLOCK NUMBER
            SA       A2,BLONUM          . RESTORE NEW BLOCK NUMBER
         L,H2    A2,PKT+3        . RETREIVE NUMBER OF WORDS ACTUALLY READ
         S       A2,BLOCK        . STORE THIS BLOCKSIZE
            LA,U     A2,8               . SET A2 = 8 SECTORS (1 BLOCK)
            AA       A2,PKT+5           . INCREMENT SECTOR NUMBER
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
            SA        A2,PKT+5          . STORE NEW SECTOR # IN PACKET
         L,S1     A2,PKT+3      . RETREIVE I/O STATUS CODE
         J        0,X2          . RETURN TO CALL LINE + 1
REWINM*
         SZ       NEED
            SZ        PKT+5     . SET NEED NEW BLOCK FLAG
            SZ        BLONUM            . RESET SECTOR # TO ZERO
         SX,H2    X11,WB+1 .              . SET ERROR BLOCK NUMBER TO ZERO
         LMJ      X11,NREWS .
         +   .    0100,(28) .
         +        $-2-DREAD,WB .
         LX,H2    X11,WB+1 .
         J        1,X11         . RETURN TO CALLING PROGRAM
           END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

ERII PROVIDE OPERATOR INTERROGATION OF PROGRAM ID/STATUS

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
$(0)      AXR$                               .
A         FORM      12,6,18                  .
ACTSWT    +         1                        . OPERATOR MESSAGE PACKET
MSPKT     +         0                        .
          +         16,$+2                   .
          +         48,OPRMSG                .
                    'ENTER MESSAGE...   '   .
                                             .
NAMSAV    +         0                        .
NEWACT    A         1,0,INTRUPT              .
          +         'OPR:  '                 .
OPRMSG    RES       8                        .
          +         '  .   '                 .
PORL      +         0                        . PROGRAM NAME PACKET
P$PKT     +         0                        .
          +         0                        .
          +         0                        .
$(1)                                         .
/.        INTER-ACTIVITY INTERRUPT SUBROUTINE
 .
 .                  CALL    ERII(PORL,LEN,RESP,ICKPT)
ERII*     .
          .         LMJ     X11,ERII
          .         +       PORL         0
          .         +       LEN          1
          .         +       RESP         2
          .         +       ICKPT        3
          .         +       (WB)         4
          .                 RETURN       4
          .
          .
          .
          LA,S1     A0,*0,X11               . A0 = P OR L PARAMETER
          TE,U      A0,'P'                  .
          SA        A0,PORL                 . ASSUME L
          LA        A0,*1,X11               . A0 = LENGTH IN WORDS
          MSI,U     A0,6                    .
          SA,H1     A0,P$PKT+1              .
          LA        A0,2,X11                .
          SA,H2     A0,P$PKT+1              .
          LA        A0,3,X11                . PICK UP ADDRESS OF ICKPT
          SA,H2     A0,CKSTOR               . STORE ADDR. OF ICKPT
          SZ        *CKSTOR                 . SET ICKPT TO ZERO
          TZ        NAMSAV                  .
          J         5,X11                   . USER IS MERELY CHANGING MESSAGE
          LA        A0,NEWACT               .
          ER        FORK$                   . INITIATE NEW ACTIVITY
          J         5,X11                   . RETURN
          .
          .
ERIIX*    SZ        ACTSWT                  . SET ACTIVITY EXIT
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
                LA       AO,NAMSAV            .
                ER       ACT$                 . DEACTIVATE THE ACTIVITY
                J        1,X11                .
        /.      SUBROUTINE INITIATED ACTIVITY
        .
        .
        INTRUPT LA,U     AO,'II'              .
                ER       NAMES$               . REQUEST NAME FOR THIS ACTIVITY
                SA       AO,NAMSAV            . SAVE NAME FOR ACT$ REQUEST
        ACTII   ER       II$                  . DEACTIVATE INTER-ACTIVITY INTERRUPT
                TNZ      ACTSWT               .
                ER       EXIT$                . EXIT THIS ACTIVITY
                SSL      AO,30                . KEEP ONLY 1ST CHARACTER OF OPERATOR MESS
                TNE,U    AO,'C'               . CHECK FOR CHECKPOINT OVERIDE
        CKSTOR  SA       AO,$-$               . SET ICKPT NON-ZERO
                TNE,U    AO,'P'               .
                J        ACTP                 .
                TE,U     AO,'M'               . OPERATOR REQUESTS PROGRAM NAME
                J        ACTII                . IGNORE INVALID REQUEST
                LA       AO,(A 1,0,0)         . OPERATOR REQUEST TO SEND MESSAGE
                SA       AO,MSPKT             . CLEAR COM$ PACKET
                LA       AO,(1,OPRMSG)        .
                LA       A1,(0,('      '))    .
                LR,U     R1,8                 .
                BT       AO,0,*A1             . CLEAR MESSAGE BUFFER
                LA,U     AO,MSPKT             .
                ER       COM$                 . PROMPT OPERATOR FOR MESSAGE
                TZ       PORL                 .
                J        ACTII                .
                LA       AO,(A 1,9,OPRMSG-1)  .
                ER       PRINT$               .
                J        ACTII                .
        ACTP    LA       AO,(A 1,1,0)         . CLEAR COM$ PACKET
                SA       AO,PSPKT             .
                LA,U     AO,PSPKT             .
                ER       COM$                 . SEND PROGRAM NAME TO OPERATOR
                J        ACTII                .
                END                           .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

FACREJ DECODE OCTAL FACILITY REJECTION CODE TO ENGLISH

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...

                 AXR$
      PRT        FORM       12,6,18 .
      $(1)        ERU$        .
      /.          'FACREJ' SUBROUTINE -- FACILITIES REJECT/WARNING PRINTOUT (BLAKLEY)
       .
       .
      FACREJ.     .         CALL 'FACREJ' USING EWORD, WB.
                  .
                  .         LMJ        X11,FACREJ
                  .         +          EWORD       0
                  .         +          WB          1
                  .                    RETURN      2
                  .
                  .
                  .
                  DS        A0,SVA0 .
                  DS        A2,SVA2 .
                  SX        X11,SVX11 .
                  LA        A3,*0,X11 .
                  SA        A3,EWORD .
                  LMJ       X11,BINOCT .
                  PRT       32,0,HEAD .
                  PRT       12,0,EWORD .
                  LA        A0,(PRT 1,8,HEAD) .
                  ER        PRINT$ .
                  LA        A3,EWORD .
                  LA,U      A1,35 .
                  JN        A3,ON .
      LOOP        LSSL      A3,1 .
      LRET        ANA,U     A1,1 .
                  JNZ       A3,LOOP .
                  DL        A0,SVA0 .
                  DL        A2,SVA2 .
                  LX        X11,SVX11 .
                  J         2,X11 .
      ON          LA        A0,TABLE,A1 .
                  JN        A0,MULLINE .
                  ER        PRINT$ .
                  J         LRET .                    SAVE A0 CTL WORD
      MULLINE     LA        A2,A0 .
      NXTLINE     LA        A0,,A2 .
                  ER        PRINT$ .
                  AT        A2,([illegible],0,1) .
                  JN        A2,[illegible]LINE .
                  J         LRET .
      $(0)
      EWORD       RES       1 .
      HEAD                  'FACILITY REQUEST STATUS WORD =
      SVA0        RES       2 .
      SVA2        RES       2 .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
SVXII     RES       1 .
/.
 .
TABLE      PRT         1, 5,BIT0 .
           PRT         1, 5,BIT1 .
           PRT         1, 5,BIT2 .
           PRT         1, 5,BIT3 .
           PRT         1, 5,BIT4 .
           PRT         1, 4,BIT5 .
           PRT         1, 9,BIT6 .
           PRT         1,11,BIT7 .
            PRT         -2,0,ABIT8 .
           PRT         1, 6,BIT9 .
           PRT         1, 9,BIT10 .
           PRT         1, 9,BIT11 .
           PRT         1, 5,BIT12 .
           PRT         1,10,BIT13 .
           PRT         1,10,BIT14 .
           PRT         1,12,BIT15 .
           PRT         1, 8,BIT16 .
            PRT         -4,0,ABIT17 .
           PRT         1, 8,BIT18 .
           PRT         1, 8,BIT19 .
           PRT         1, 8,BIT20 .
           PRT         1,13,BIT21 .
           PRT         1, 7,BIT22 .
           PRT         1, 7,BIT23 .
           PRT         1, 5,BIT24 .
           PRT         1, 5,BIT25 .
           PRT         1, 5,BIT26 .
           PRT         1, 5,BIT27 .
           PRT         1, 3,BIT28 .
           PRT         1, 6,BIT29 .
           PRT         1, 3,BIT30 .
           PRT         1,10,BIT31 .
           PRT         1, 7,BIT32 .
           PRT         1, 6,BIT33 .
            PRT         -3,0,ABIT34 .
           PRT         1, 5,BIT35 .
BIT0     'BIT 0  RESERVED FOR FUTURE USE'
BIT1     'BIT 1  RESERVED FOR FUTURE USE'
BIT2     'BIT 2  RESERVED FOR FUTURE USE'
BIT3     'BIT 3  RESERVED FOR FUTURE USE'
BIT4     'BIT 4  RESERVED FOR FUTURE USE'
BIT5     'BIT 5  F-CYCLE CONFLICT '
BIT6     'BIT 6  FILE DISABLED,  UNRECOVERABLE ROLLED OUT FILE  '
BIT7     'BIT 7  FILE DISABLED,  FILE ASSIGNED WRITE-ENABLED DURING RECOVERY'
BIT8A     'BIT 8  FILE DISABLED, LINKS PERTINENT TO MASTER FILE DIRECTORY
BIT8B     '        ITEMS HAVE BEEN DESTROYED
BIT9     'BIT 9  REQUESTED EQUIPMENT IS DOWN
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
BIT10     'BIT 10 WRITE-ONLY FILE CATALOGUED WITH A (W) OPTION    '
BIT11     'BIT 11 READ-ONLY FILE CATALOGUED WITH A (R) OPTION    '
BIT12     'BIT 12 EQUIPMENT IS TAPE       '
BIT13     'BIT 13 PROJECT-ID INCORRECT FOR CATALOGUED PRIVATE FILE      '
BIT14     'BIT 14 FILE TO BE DECATALOGUED WHEN NO RUN HAS FILE ASSIGNED'
BIT15     'BIT 15 REQUESTED CATALOGUED FILE WAS ALREADY ASSIGNED TO ANOTHER R'
          'UN    '
BIT16     'BIT 16 FILE ASSIGNED EXCLUSIVELY TO ANOTHER RUN '
BIT17A     'BIT 17 OPTION CONFLICT FOR CATALOGUED FILES, EITHER THE D    '
BIT17B     '        AND K OPTIONS WERE SPECIFIED OR C OR U, OR P, R OR W '
BIT17C     '        IN COMBINATION WITH C OR U WAS SPECIFIED FOR A FILE  '
BIT17D     '        WHICH ALREADY EXISTS IN THE DIRECTORY.    ' .
BIT18     'BIT 18 REQUEST ON WAIT STATUS FOR FACILITIES      '
BIT19     'BIT 19 MASS STORAGE FILE HAS BEEN ROLLED OUT      '
BIT20     'BIT 20 INVALID REEL NR OR PACK-ID FOR CAT FILE  '
BIT21      'BIT 21 (A) OPTION SPECIFIED & FILENAME NOT FOUND IN MASTER FILE DI'
           'RECTORY      ' .
BIT22     'BIT 22 WRITE KEY SPECIFIED, NONE EXISTS    '
BIT23     'BIT 23 READ KEY SPECIFIED, NONE EXISTS    '
BIT24     'BIT 24 READ KEY UNSPECIFIED  '
BIT25     'BIT 25 WRITE KEY UNSPECIFIED   '
BIT26     'BIT 26 INCORRECT WRITE KEY      '
BIT27     'BIT 27 INCORRECT READ KEY       '
BIT28     'BIT 28 NOT USED    '
BIT29     'BIT 29 INTERNAL FILENAME NOT UNIQUE '
BIT30     'BIT 30 NOT USED    '
BIT31      'BIT 31 EQUIPMENT TYPE NOT COMPATABLE WITH CATALOGUED TYPE    '
BIT32     'BIT 32 FILE WAS PREVIOUSLY CATALOGUED         '
BIT33      'BIT 33 FILE WAS PREVIOUSLY ASSIGNED    ' .
BIT34A     'BIT 34 FIELD ERROR IN CONTROL STATEMENT OTHER THAN SYNTAX,  '
BIT34B     '        OPTION CONFLICT, OR                                  '
BIT34C     '        REQUESTED HARDWARE NOT PART OF PRESENT SYSTEM        '
BIT35     'BIT 35 REQUEST NOT ACCEPTED    '
ABIT8       PRT       1,11,BIT8A .
            PRT       1, 6,BIT8B .
ABIT34      PRT       1,10,BIT34A .
            PRT       1,10,BIT34B .
            PRT       1,10,BIT34C .
ABIT17      PRT       1,10,BIT17A .
            PRT       1,10,BIT17B .
            PRT       1,10,BIT17C .
            PRT       1, 8,BIT17D .
 /.         LITERALS
  .
            END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE FMTGEN(LOPT)
C***** IHISTB - TABLE CONTAINING LENGTH AND POSITION  OF HIS. VARIABLES
C      ITSKTB - TABLE CONTAINING RELATIVE POSITION OF TASK VARIABLES
C      ICHKTB - TABLE CONTAINING RELATIVE POSITION OF CHECK VARIABLES
C***** ICHEK - ARRAY CONTAINING VALUE OF VARIABLES THAT MUST MATCH
C      IFMT - GENERATED FORMAT ARRAY
C***** ICHAR - INPUT FORMAT CONTROL CHARACTERS
C      NCARD - TOTAL NO. OF CARDS
C***** M - NO. OF CARDS PROCESSED THUS FAR
C***** KTH - COUNTS HISTORY VARIABLES
C***** KTT - COUNTS TASK VARIABLES
C***** KTC - COUNTS CHECKS
C***** NCHEK - TOTAL NO. OF CHECKS
C***** NR - NO. OF REPEATED FORMAT TYPES
C***** IAXC - CURRENT A OR X
C***** IAXP - PREVIOUS A OR X
C***** LAC - CURRENT LENGTH OF A FIELD
C***** LAP - PREVIOUS LENGTH OF A FIELD
C***** KTCR - COUNTS CHARACTERS AS THEY ARE SCANNED
C***** KTA - COUNTS A FORMATS
C***** KTFC - COUNTS NO. CHAR. USED IN CURRENT FORMAT WORD
C***** KTFW - COUNTS FORMAT WORDS THAT ARE USED
C***** NFC - NO. OF CHAR. TO INSERT IN FORMAT
C***** LFW - LOC. OF WORD CONTAINING FMT. CHAR. LF. ADJ.
C***** ICMA - INSERT COMMA OR OPEN PAREN.
C***** KAR - CURRENT CHAR.
C***** KARP - PREVIOUS CHAR.
C***** LHV - LENGTH OF HISTORY VAR. IN CHAR.
C***** LFWTB - TABLE OF LAST FORMAT WORD FOR EACH CARD
C***** NTHTB - TABLE OF NO. OF THINGS TO READ FROM EACH CARD
      COMMON/FGEN/IFMT(1000), ITSKTB(1700), IHISTB(1700), ICHKTB(500),
     1ICHEK(500), LFWTB( 50), NTHTB( 50), NCHEK, NCARD, KTH, KTT
      DIMENSION ICHAR(80)
      INTEGER C,T,H,P,B,A,X,TLCHEK
      DATA INX/1HI/,A/1HA/,X/1HX/,C/1HC/,T/1HT/,H/1HH/,P/1H./,B/1H /,
     1IOPEN,ICLOSE,ICLOPT/'(',')',',A4'/
      ASSIGN 127 TO TLCHEK
      IF(LOPT.EQ.3) ASSIGN 132 TO TLCHEK
      ASSIGN 155 TO MAD
      ASSIGN 63 TO MUG
      MFN = 1
      IF(LOPT.NE.2) GO TO 45
      ASSIGN 144 TO MAD
      ASSIGN 62 TO MUG
      MFN = 3
   45 CONTINUE
      MOV3=2**18
      NONE = 1
      NAMI = 1700
      NATA = 1700
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      KTFC=0
      KTFW=1
      ICMA=IOPEN
      KTCR=0
      KTM=1
      KTA=0
      KTC=0
      KTT=0
      IHISTB(1)=4*MOV3+1
      DO 50 I=2,1000
   50 IFMT(I)=0
C***** READ FORMAT CONTROL CHAR. FOR ONE CARD
      READ 953, (ICHAR(I),I=1,80)
  953 FORMAT(80A1)
C***** LOOP FOR CHARACTERS
      M=0
   55 KTCR=KTCR+1
C***** TEST FOR FIRST CARD
      IF(KTCR.EQ.1) GO TO 65
C***** TEST FOR NEW CARD
      IF(KTCR.LE.80) GO TO 85
      M=M+1
C***** INSERT PREVIOUS FORMAT
      CALL CPI(NR,IAXP,LAP,LFW,NFC,KTFW,KTFC,ICMA,LOPT)
C       IF LAST CHAR. WAS AN H OR P OR T, INSERT CURRENT FORMAT
        IF(LOPT.NE.3.AND.KAR.NE.H.AND.KAR.NE.P.AND.KAR.NE.T) GO TO 60
      IF(LOPT.EQ.3 .AND. KAR.NE.'F' .AND. KAR.NE.'X' .AND. KAR.NE.H
     1  .AND. KAR.NE.P .AND. KAR.NE.T) GO TO 60
      NR=1
      CALL CPI(NR,IAXC,LAC,LFW,NFC,KTFW,KTFC,ICMA,LOPT)
      IF (LOPT.NE.3.AND.IAXC.EQ.INX.AND.LAC.NE.LTF) GO TO 143
      IF(IAXC.EQ.INX .OR. IAXC.EQ.'F') GO TO 60
C***** STORE LENGTH IN HISTORY TABLE
      IHISTB(KTM)=IHISTB(KTM)+LHV*MOV3
C***** CLOSE OUT FORMAT FOR THIS CARD
   60 GO TO MUG,(62,63)
   62 KTA = KTA + 1
      CALL INSERT(KTFW,KTFC,ICLOPT,MFN)
   63 LFWTB(M) = KTFW
      NTHTB(M) = KTA
      CALL INSERT(KTFW,KTFC,ICLOSE,NONE)
C***** TEST FOR END OF FORMAT CONTROL CHARACTERS
      IF(M.GE.NCARD) GO TO 170
C***** BEGIN FORMAT FOR NEW CARD
      IF(KTFC.NE.0) KTFW=KTFW+1
      KTFC=0
      ICMA=IOPEN
C***** READ FORMAT CONTROL CHAR. FOR ONE CARD
      READ 953, (ICHAR(I),I=1,80)
      KTCR=1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
C***** TEST IF FIRST 4 CHAR. ON CARD = C...
 65    IF(ICHAR(KTCR).NE.C) GO TO 75
       DO 70 I=1,3
       KTCRI=KTCR+1
 70    IF(ICHAR(KTCRI).NE.P) GO TO 75
       IF(ICHAR(KTCR+4).NE.P) GO TO 80
 75    PRINT 954
  954 FORMAT(54H  FORMAT CONTROL CARD DOES NOT BEGIN C... RUN ABORTED )
       CALL ERTRAN(2)
 80    NR=1
       IAXP=A
       LAP=4
       KARP=C
       KTCR=KTCR+4
       KTA=KTA+1
C***** ANALYZE CURRENT CHARACTER
 85    KAR=ICHAR(KTCR)
C***** TEST FOR ,
       IF(KAR.EQ.P) GO TO 110
C***** TEST FOR PREVIOUS H
       IF(KARP.EQ.H) GO TO 130
C       TEST FOR PREVIOUS T
       IF (KARP.EQ.T) GO TO TLCHEK(127,130,132)
       IF (KARP.EQ.'F' .OR. KARP.EQ.'X') GO TO 132
C***** TEST FOR H
 90    IF(KAR.EQ.H) GO TO 145
C***** TEST FOR BLANK
       IF(KAR.EQ.B) GO TO MAD,(155,144)
C***** T OR OTHER - A1 FORMAT
       KTA=KTA+1
       IAXC=A
       LAC=1
C***** TEST FOR T
       IF(LOPT.EQ.3 .AND.(KAR.EQ.T .OR. KAR.EQ.'F' .OR. KAR.EQ.'X')) GO
     1 TO 160
       IF(KAR.EQ.T) GO TO 160
C***** OTHER - MAKE ENTRY IN CHECK TABLE AND STORE CHECK VALUE
       KTC=KTC+1
       ICHKTB(KTC)=KTA
       ICHEK(KTC)=KAR
C***** TEST FOR TOO MANY CHECK CHARACTERS
       IF(KTC.LE.500) GO TO 95
       PRINT 955
 955   FORMAT('  THE NUMBER OF FORMAT CARD CHECK CHARACTERS EXCEEDS 500.
     1RUN ABORTED.')
       CALL ERTRAN(2)
C***** COMPARE CURRENT FORMAT WITH PREVIOUS FORMAT
 95    IF(IAXC.NE.IAXP) GO TO 100
       IF(IAXC.EQ.A.AND.LAC.NE.LAP)GO TO 100
C***** SAME FORMAT
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      NR=NR+1
      GO TO 105
C***** DIFFERENT FORMATS - INSERT PREVIOUS ONE
100   CALL CPI (NR,IAXP,LAP,LFW,NFC,KTFW,KTFC,ICMA,LOPT)
      NR=1
      IAXP=IAXC
      LAP=LAC
105   KARP=KAR
C***** THROUGH ANALYSING THIS CHARACTER
      GO TO 55
C  DOT - TEST FOR PREVIOUS H, DOT, OR T
 110  IF(KARP.EQ.H.OR.KARP.EQ.P.OR.KARP.EQ.T)GO TO 115
      IF(LOPT.EQ.3 .AND. (KARP.EQ.'F' .OR. KARP.EQ.'X')) GO TO 115
      PRINT 960
960   FORMAT('  A FORMAT CARD DOT (.) WAS USED INCORRECTLY. RUN ABORTED.
     1')
      CALL ERTRAN(2)
C  DOT PRECEDED BY H, DOT, OR T - INCREMENT LENGTH
115   LAC=LAC+1
      LHV=LHV+1
C***** TEST FOR LENGTH OF A-FIELD .GT. 6
      IF(LAC.LE.6) GO TO 55
C***** CHANGE A7 TO A6,A1
      KTA=KTA+1
      LAC=6
C***** COMPARE PREVIOUS FMT WITH CURRENT FMT
      IF(IAXC.NE.IAXP) GO TO 120
      IF(LAC.NE.LAP) GO TO 120
      NR=NR+1
      GO TO 125
120   CALL CPI (NR,IAXP,LAP,LFW,NFC,KTFW,KTFC,ICMA,LOPT)
      NR=1
      IAXP=IAXC
      LAP=LAC
125   LAC=1
      GO TO 55
C     SET TASK FIELD LENGTH COMPARITOR
127   ASSIGN 130 TO TLCHEK
      LTF = LAC
C     TEST TASK FIELD LENGTH FOR CONSISTENCY
130   IF (IAXC.EQ.INX.AND.LAC.NE.LTF) GO TO 143
C  NOT-DOT PRECEDED BY H, DOT, OR T - COMPARE PREV. WITH CURR.
132   IF (IAXC.NE.IAXP) GO TO 135
      IF(LAC.NE.LAP) GO TO 135
      NR=NR+1
      GO TO 140
135   CALL CPI (NR,IAXP,LAP,LFW,NFC,KTFW,KTFC,ICMA,LOPT)
      NR=1
      IAXP=IAXC
      LAP=LAC
```

LINE # TEXT...

```
C     STORE LENGTH IN HISTORY TABLE IF HISTORY VARIABLE
 140  IF(IAXP.EQ.INX .OR. IAXP.EQ.'F') GO TO 90
       IHISTB(KTH)=IHISTB(KTH)+LHV*MOV3
      GO TO 90
143   WRITE (6,943)
943   FORMAT ('  MORE THAN ONE TASK LENGTH ON FORMAT CARDS. RUN ABORTED.
     1')
      CALL ERTRAN(2)
144   CONTINUE
C  LOPT=2 PROCESSOR
      IAXC = X
      KTA = KTA + 1
      LAC = 1
      GO TO 95
C***** H - MAKE ENTRY IN HISTORY TABLE
 145  KTH=KTH+1
      IF(KTH.LE.NAHI) GO TO 148
      PRINT 963,NAHI
963   FORMAT('  THE NUMBER OF HISTORY VARIABLES DEFINED BY FORMAT CARDS
     1EXCEEDS ',I4,'.  RUN ABORTED.')
      CALL ERTRAN(2)
148   KTA = KTA + 1
      IHISTB(KTH)=KTA
      IAXC=A
      LAC=1
      LHV=1
      GO TO 105
C***** BLANK - X FORMAT
 155  IAXC=X
      GO TO 95
C***** T - MAKE ENTRY IN TASK TABLE
 160  KTT=KTT+1
      IF(KTT.LE.NATA) GO TO 165
      PRINT 965,NATA
965   FORMAT('  THE NUMBER OF TASK VARIABLES DEFINED BY FORMAT CARDS EXC
     1EEDS ',I4,'.  RUN ABORTED.')
      CALL ERTRAN(2)
165   ITSKTB(KTT) = KTA
       IAXC=INX
      IF(KAR.EQ.'F' .OR. KAR.EQ.'X') IAXC = 'F'
       LAC=1
       LHV=1
      GO TO 105
 170  NCHEK=KTC
       RETURN
       END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE FORMAT(LDICT,IMAT,NCHAR,ITYPE,IFLD1,IFLD2,$)
      DIMENSION LDICT(2), IMAT(2), LIST(10)
      NCHAR = 0
      IFLD1 = 0
      IFLD2 = 0
      ITYPE = -1
      IMAT(2) = ' '
      IF(LDICT(2).EQ.' ') GO TO 900
      ISW = 0
      DO 3 IWORD = 2, 1, -1
      DO 2 ICHAR = 6, 1, -1
      IBIT = IABS(ICHAR*6 - 6)
      JCHAR = FLD(IBIT,6,LDICT(IWORD))
      IF(ISW.NE.0) GO TO 1
      IF(JCHAR.EQ.1R ) GO TO 2
      NCHAR = NCHAR + 1
      DECODE(100,JCHAR) LIST(NCHAR)
      ISW = 1
      GO TO 2
 1    CONTINUE
      IF(JCHAR.EQ.1R ) GO TO 4
      NCHAR = NCHAR + 1
      DECODE(100,JCHAR) LIST(NCHAR)
 100  FORMAT(5X,A1)
      IF(JCHAR.EQ.1RA) ITYPE = -1
      IF(JCHAR.EQ.1RI) ITYPE =  0
      IF(JCHAR.EQ.1RF) ITYPE = +1
 2    CONTINUE
 3    CONTINUE
 4    CONTINUE
      ENCODE(102,IMAT) (LIST(I),I=NCHAR,1,-1)
 102  FORMAT(12A1)
      IF(ITYPE) 20, 30, 40
 20   CONTINUE
      IF(NCHAR.LE.3) GO TO 25
      ICHAR = NCHAR
      CALL BCDBIN(LIST(ICHAR),IFLD1,1,$901)
      ICHAR = NCHAR - 1
      CALL BCDBIN(LIST(ICHAR),NUM,1,$25)
      IFLD1 = IFLD1*10 + NUM
 25   CONTINUE
      CALL BCDBIN(LIST(1),IFLD2,1,$901)
      RETURN
 30   CONTINUE
      CALL BCDBIN(LIST(1),IFLD1,1,$901)
      RETURN
 40   CONTINUE
      CALL BCDBIN(LIST(1),IFLD2,1,$901)
      CALL BCDBIN(LIST(3),IFLD1,1,$901)
      CALL BCDBIN(LIST(4),NUM,1,$42)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
       IFLD1 = NUM*10 + IFLD1
 42    RETURN
 900   CONTINUE
       ENCODE(103,IMAT)
 103   FORMAT(' NO FORMAT')
       RETURN 7
 901   CONTINUE
       ENCODE(104,IMAT)
 104   FORMAT(' INVALID  ')
       RETURN 7
       END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
$(0)    AXR$ .
SV      RES      1 .
$(1)    ERU$ .
FREDEV*                          . ENTRY POINT FOR FREE DEVICE FOR FORTRAN
        SA      A0,SV            . SAVE REGISTER A0
        LA      A0,*0,X11        . RETREIVE FORTRAN UNIT NUMBER
        SZ      NTAB$,A0         . ZERO OUT NTAB$ ENTRY SO FORTRAN 'FORGETS'
        LA      A0,SV            . RESTORE REGISTER A0
        J       2,X11            . RETURN TO USER
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE FREE(IUNIT)
      DIMENSION IMAGE(2)
      ENCODE(100,IMAGE) IUNIT
100   FORMAT('WFREE ',I2,'. . ')
      CALL ERTMAN(6,IMAGE)
      CALL FREDEV(IUNIT)
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

FREES FREE FILE BUT RETAIN TAPE DRIVE FOR ANOTHER ASSIGNMENT

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE FREES(IUNIT)
      DIMENSION IMAGE(3)
      ENCODE(100,IMAGE) IUNIT
100   FORMAT('wFREE,5 ',I2,'. . ')
      CALL ERTRAN(6,IMAGE)
          CALL DELAY(7)
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE FSORT(IUNIT,LRL,ITYPE,IORDER)
C  IUNIT = FORTRAN UNIT WITH UNFORMATTED RECORDS EACH OF LENGTH
C  "LRL" WHERE LRL<132.  THE FIRST WORD MUST BE THE SORT FIELD.
C  ITYPE = SORT TYPE (0=ALPHANUMERIC,1=NUMERIC)
C  IORDER = SORT ORDER (0=ASCENDING,1=DESCENDING)
C    RECORDS ARE READ FROM IUNIT, SORTED, AND RE-WRITTEN TO IUNIT.
      INTEGER TBL
      DIMENSION TBL(13), IBUF(132)
      DATA TBL/1,1,36,0,0,1,99999,1,36,0,0,2,99999/
          MAXLEN = 1
101   IF(LRL.LE.132) GO TO 199
      WRITE(6,100) LRL
100   FORMAT(' LRL SPECIFIED = ',I6,'.  MAX ALLOWED = 132.  RUN HALTS.')
      CALL ERTRAN(2)
199   TBL(4) = ITYPE
      TBL(5) = IORDER
      REWIND IUNIT
      CALL SOPEN3($200,$300,LRL,MAXLEN,TBL)
200   READ(IUNIT,END=250) (IBUF(I),I=1,LRL)
      CALL SRREL(IBUF,LRL)
      GO TO 200
250   CALL SSORT
300   REWIND IUNIT
301   CALL SRRET(IBUF,N,$350)
      WRITE(IUNIT) (IBUF(I),I=1,N)
      GO TO 301
350   ENDFILE IUNIT
      REWIND IUNIT
      RETURN
      ENTRY FSORT2(IUNIT,LRL,ITYPE,IORDER,JTYPE,JORDER)
          MAXLEN = 2
          TBL(7) = 2
          TBL(10) = JTYPE
          TBL(11) = JORDER
          GO TO 101
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE GETFIL(IMAGE)
C THIS SUBROUTINE WILL CONTINUE TO REQUEST A FILE UNTIL
C        IT BECOMES AVAILABLE.
      DIMENSION IMAGE(14), MESOUT(8)
C  MSKWOF IS A MASK FOR A "WAIT ON FACILITIES" FAC REJECT CODE
      DATA MSKWOF/0000001000000/
      LENR = 1
3     LOOP = 0
1     CONTINUE
      ISTAT = NERTRN(6,IMAGE)
      IF(ISTAT.GE.0) RETURN
      JSTAT = AND(ISTAT,MSKWOF)
      IF(JSTAT.EQ.0) GO TO 2
      LOOP = LOOP + 1
      IF(LOOP.GE.8) GO TO 4
      CALL DELAY(15)
      GO TO 1
2     CALL FACREJ(ISTAT)
      STOP
4     ENCODE(100,MESOUT)
100   FORMAT('WAITING FOR TAPE DRIVE.    AE')
      CALL OMSGW(5,MESOUT,LENR,MESIN)
      DECODE(101,MESIN) IANS
101   FORMAT(A1)
      IF(IANS.NE.'E') GO TO 3
      ENCODE(102,MESOUT)
102   FORMAT('DO YOU REALLY INTEND TO ERROR THIS RUN?   YES/NO')
      CALL OMSGW(8,MESOUT,LENR,MESIN)
      DECODE(101,MESIN) IANS
      IF(IANS.NE.'Y') GO TO 3
      GO TO 2
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE GETMAS(IMAGE)
C THIS SUBROUTINE WILL CONTINUE TO REQUEST A FILE UNTIL
C       IT BECOMES AVAILABLE.
      DIMENSION IMAGE(14)
C  MSKWOF IS A MASK FOR A "WAIT ON FACILITIES" FAC REJECT CODE
      DATA MSKWOF/000000100000O/
 1    CONTINUE
      ISTAT = NERTRN(6,IMAGE)
      IF(ISTAT.GE.0) RETURN
      JSTAT = AND(ISTAT,MSKWOF)
      IF(JSTAT.EQ.0) GO TO 2
      CALL DELAY(30)
      GO TO 1
 2    WRITE(6,300) (IMAGE(I),I=1,5)
 300  FORMAT(' ASSIGNMENT REJECTED = ',5A6)
      CALL FACREJ(ISTAT)
      CALL ERTRAN(2)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
$(0)       AXR$                    .
SVA        RES      2              .
IRET       RES      1              .
$(1)       EXUS                    .
GETPCD*    DS       A0,SVA         . SAVE A0,A1
           LA       A1,*0,X11      . RETREIVE RELATIVE LOCATION (0-03507)
           AA       A1,(1,0)
           LA,U     A0,0           . ZERO OUT A0
           LA       A0,1,X11       . RETREIVE OUTPUT ADDRESS
           ER PCTS                 . CALL PCTS
           DL       A0,SVA         . RESTORE A0,A1 REGISTERS
           J        3,X11          . RETURN TO CALLER
           END
```

GETPCT RETREIVE INFORMATION FROM THE PROGRAM CONTROL TABLE (OCT)

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE GETPCT(IOCT,LOC)
      ENCODE(100,IDEC) IOCT
100   FORMAT(I4)
      DECODE(101,IDEC) IOCTT
101   FORMAT(O4)
      CALL GETPCD(IOCTT,LOC)
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE H2ACND(NNDUP,LPACK)
C THIS SUBROUTINE WILL ACCEPT THE PACKED/SPLIT WORD FORMAT FOR
C THE OPTRAN DUP. SHEET POINTERS AND SORT THEM INTO ASCENDING
C SEQUENCE ON THE FRONTPAGE CODES IN THE RIGHT HALF (H2) OF THE
C PACKED WORD FORMAT.  THE NUMBER OF NET DUPS FOR THIS CASE
C AND THE PACKED POINTER ARRAY ARE THE INPUT. FOR OUTPUT, THE
C SUBROUTINE SIMPLY RESEQUENCES THE PACKED FORMAT ARRAY INTO THE
C PROPER ORDER.  THIS VERSION IS AN EDITED COPY OF H2SORT.
      DIMENSION LPACK(1)
      DO 3 IRESP = 2, NNDUP
      I = IRESP - 1
      J = IRESP
2     CONTINUE
      IF(FLD(18,18,LPACK(J)).GE.FLD(18,18,LPACK(I))) GO TO 3
      IHOLD = LPACK(J)
      LPACK(J) = LPACK(I)
      LPACK(I) = IHOLD
      J = I
      I = I - 1
      IF(I.GT.0) GO TO 2
3     CONTINUE
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      SUBROUTINE H2SORT(NRESP,LPACK)
C THIS SUBROUTINE WILL ACCEPT THE PACKED/SPLIT WORD FORMAT FOR
C THE INDIVIDUAL'S TASK RESPONSES AND SORT THEM INTO DESCENDING
C SEQUENCE ON THE TASK RESPONSES IN THE RIGHT HALF (H2) OF THE
C PACKED WORD FORMAT.  THE NUMBER OF RESPONSES FOR THIS INDIVIDUAL
C AND HIS PACKED RESPONSE ARRAY ARE THE INPUT. FOR OUTPUT, THE
C SUBROUTINE SIMPLY RESEQUENCES THE PACKED FORMAT ARRAY INTO THE
C PROPER ORDER.
      DIMENSION LPACK(1)
      DO 3 IRESP = 2, NRESP
      I = IRESP - 1
      J = IRESP
    2 CONTINUE
      IF(FLD(18,18,LPACK(J)).LE.FLD(18,18,LPACK(I))) GO TO 3
      IHOLD = LPACK(J)
      LPACK(J) = LPACK(I)
      LPACK(I) = IHOLD
      J = I
      I = I - 1
      IF(I.GT.0) GO TO 2
    3 CONTINUE
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE INSERT(KTFW,KTFC,LFW,NFC)
      COMMON/FGEN/IFMT(1000), DUMMY(4504)
C  THIS ROUTINE PACKS THE LEFTMOST NFC CHARACTER POSITIONS OF THE WORD
C  LFW INTO THE NEXT AVAILABLE POSITIONS OF THE ARRAY IFMT.
C  DEFINITIONS -
C                  IMAT = VARIABLE FORMAT ARRAY USED FOR ENCODING (AN,A6)
C                  KTFC = LAST CHARACTER NO. INSERTED (0-5)
C                  KTFW = SUBSCRIPT OF IFMT IN WHICH TO STORE
C                  LFW  = WORD TO BE INSERTED
C                  NFC  = NO. OF CHARACTERS TO BE INSERTED
C
      DIMENSION IMAT(2)
      IF(KTFC.EQ.0) GO TO 4
      ENCODE(100,IMAT) KTFC
  100 FORMAT(2H(A,I1,4H,A6) )
      IWORD = IFMT(KTFW)
      ENCODE(IMAT,IFMT(KTFW)) IWORD, LFW
    1 NA = 6 - KTFC
      IF(NA - NFC) 2,3,5
    2 KTFC = NFC - NA
      KTFW = KTFW + 1
      IF(KTFW.LE.1000) RETURN
      GO TO 6
    3 KTFC = 0
      KTFW = KTFW + 1
      IF(KTFW.LE.1000) RETURN
      GO TO 6
    4 IFMT(KTFW) = LFW
    5 KTFC = KTFC + NFC
      RETURN
    6 WRITE(6,101)
  101 FORMAT('  THE FORTRAN FORMAT GENERATED FROM FORMAT CARDS EXCEEDS A
     1LLOCATED STORAGE. SIMPLIFY INPUT. RUN ABORTED.')
       STOP
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE IRANF
      CALL ERTRAN(9,IDATE,ITIME)
      IMATCH = AND(IDATE,ITIME)
      ISEED = IMATCH + ITIME + 1
      WRITE(6,100) ISEED
100   FORMAT(1X,'SELECTED SEED = ',O12)
      RETURN
      ENTRY GETRAN(ANS)
      CALL NRAND(ISEED,ANS)
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE JDCOPY
      PARAMETER NTDM = 1750
      DIMENSION AK(NTDM), AVE(NTDM), ANUMP(NTDM), JA(50), NMASK(600)
      LAST = 0
      REWIND 25
      REWIND 27
 1    READ(27,END=2) IDSP,NTD,(AK(K),K=1,NTD),NC35,(NMASK(K),K=1,NC35),
     1JSEQ,IR,(JA(K),K=1,IR),(AVE(K),K=1,NTD),(ANUMP(K),K=1,NTD)
      IF(JSEQ.LE.LAST) GO TO 3
      LAST = JSEQ
      WRITE(25)      IDSP,NTD,(AK(K),K=1,NTD),NC35,(NMASK(K),K=1,NC35),
     1JSEQ,IR,(JA(K),K=1,IR),(AVE(K),K=1,NTD),(ANUMP(K),K=1,NTD)
      GO TO 1
 2    RETURN
 3    WRITE(6,100) JSEQ, LAST
 100  FORMAT('1FILE ERROR REPORT NUMBER ',I6,' FOLLOWS ',I6,' ON JOB DES
     1CRIPTION FILE.  RUN ABORTS.')
      CALL ERTRAN(2)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE JDEOF
         IUNIT = 25
      REWIND IUNIT
 1    READ(IUNIT,END=2) KILL
      GO TO 1
 2    BACKSPACE IUNIT
      RETURN
      END
```

LASCMP LOGICAL ALPHANUMERIC STRING COMPARISON

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      FUNCTION LASCMP(IV,LO,IH,N)
      DIMENSION IV(1), LO(1), IH(1)
      IF(N.GT.0) GO TO 2
    1 LASCMP = 0
      RETURN
    2 ASSIGN 3 TO JUMP
      IST = 0
      DO 8 IWORD = 1, N
      DO 7 IHALF = 1, 2
      LOW = FLD(IST,18,LO(IWORD))
      IHI = FLD(IST,18,IH(IWORD))
      IVL = FLD(IST,18,IV(IWORD))
      GO TO JUMP, (3,5,6)
    3 IF(IVL.GT.IHI.OR.IVL.LT.LOW) GO TO 1
      IF(IVL.NE.IHI.AND.IVL.NE.LOW) GO TO 9
      IF(IVL.EQ.IHI) GO TO 4
      ASSIGN 5 TO JUMP
      GO TO 7
    4 IF(IVL.EQ.LOW) GO TO 7
      ASSIGN 6 TO JUMP
      GO TO 7
    5 IF(IVL.GE.LOW) GO TO 7
      GO TO 1
    6 IF(IVL.GT.IHI) GO TO 1
    7 IST = 18 - IST
    8 CONTINUE
    9 LASCMP = 1
      RETIRN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

```
LINE #   TEXT...

 . ****************************************************************************2
 .                                                                            3
 . PARAMETER TABLE FOR SORT SUBROUTINE                                        4
 .                                                                            4
 $(0).                                                                        5
 F          FORM      6,30.                                                   6
 F1         FORM      6,6,6,6,12.                                             7
 F2         FORM      6,12,18.                                                8
 P        FORM     6,6,6,18                                                   9
 TAG1     P        0,1,3,MESS87
 TAG2     P        0,1,9,KEYMSG
 TAG3     P        0,1,6,ARGMSG
 TABLE       F          0,TABLE.             TABLE HEADER WORD                10
             F          01,0.                MAXIMUM NO. OF WORDS IN RECORD   11.1
             F          02,0.                LINK SIZE, IN WORDS, OF THE RECORDS  11
             F          042,0.               RETURN ADDRESS FROM SOPENX  - FPOC   12
             F          043,0.               RETURN ADDRESS FROM SSORT   - LPOC   13
             RES        100.                 AREA FOR I/O AND KEYS (VARIABLE)     14
 .           F1         03,0,0,0,0.          KEY FORMAT - TYPE, ORDER, BIT-POS,   15
 .           F2         077,0,0.               FIELD-NO, WORD-POS, NO-OF-BITS     16
 SENTNL      F2         071,0,TABLE          SENTINEL SPECIFYING END OF TABLE     17
 ERRCHK      RES        6          . REGISTER SAVE AREA          *6-11-68WRT
 KEYMSG      '6 PARAMETERS HAVE NOT BEEN SPECIFIED FOR ALL SORT KEYS'
 ARGMSG      'THREE ARGUEMENTS NOT GIVEN IN CALL'   .            *6-11-68WRT
 . ****************************************************************************18
 .                                                                            19
 .                                                                            20
 . ROUTINE TO BUILD THE INTERNAL SORT PARAMETER TABLE.  UPON ENTRY, THE       21
 .   I/O TABLE ADDRESS IS IN REGISTER 12 AND THE LENGTH IS IN 65.  THE        22
 . STARTING ADDRESS OF THE USER'S PARAMETERS IS IN REGISTER 11.               23
 .       CALL SOPENX(FPOC, LPOC, MAX-NO-WORDS-IN-RECORD, NO-WORDS-IN-LINK,    23.1
 .            KEY1, KEY2, ..., KEY40, SENTINEL)                               23.2
 .       KEY FORMAT IS  WORD-POS, BIT-POS, NO-OF-BITS, TYPE, ORDER, FIELD-NO  23.3
 .       SENTINEL IS  99999                                                   23.4
 .                                                                            24
 .                                                                            25
 . SET THE TABLE ENTRIES FOR FPOC, LPOC, MAXIMUM RECORD SIZE, LINK SIZE,      26
 . AND INPUT/OUTPUT                                                           26.1
 .                                                                            27
 $(1).                                                                        28
 SBUILD*    SX         11,ERRCHK+5 . SAVE LINK ADDRESS          *6-11-68WRT
            LX1,016    11,1.                                    *6-11-68WRT
            LA,1       13,0,*11.          FPOC                                30
            SA,1       13,TABLE+3.                                            31
            LA,1       13,0,*11.          LPOC                                32
            SA,1       13,TABLE+4.                                            33
            LA         13,*0,*11.         MAXIMUM WORDS IN RECORD             33.1
            SA,1       13,TABLE+1.                                            33.2
            LA         13,*0,*11.         LINK SIZE IN WORDS                  34
            SA,1       13,TABLE+2.                                            35
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...
             LXI,14    12,1.              INPUT/OUTPUT TABLE ENTRIES            36
             LX,14     13,TABLE+5.          SPECIFIED BY USER'S CALL TO SOPENX  37
             LXI,14    13,1.                REGISTERS 12 AND 65 SET PRIOR       38
             BT        13,0,*12.              TO ENTRY                          39
             SZ        COUNT.                                                   39.1
              SX        12,ERRCHK+1 .                              *6-11-68WRT
              DS        13,ERRCHK+2                                *6-11-68WRT
              SX        15,ERRCHK+4                                *6-11-68WRT
              LX        14,(1,0)  . INITIALIZE COUNT FOR KEY PARAM  *7-31-68WRT
              LX        12,(99999).  LOAD SENTINEL                   *7-31-68WRT
              LX        11,0,11   . LOAD ADDRESS OF KEY DATA AREA     *7-05-68WRT
              LXI,016   11,1       .  LOAD INCREMENT                 *8-05-68WRT
              SX        11,ERRCHK  .  SAVE STARTING ADD. OF KEY DATA*7-31-68WRT
   LOOP1      TNE       12,0,*11  .  CHECK FOR SENTINEL WORD         *7-31-68WRT
              J         KEYCHK  .  SENTINEL DETECTED                 *7-31-68WRT
              TLEM      14,(240)  . CHECK LIMIT OF FORTY KEYS        *7-31-68WRT
              J         LOOP1  . LESS THAN 40 - CONTINUE SEARCH      *7-31-68WRT
           LA       12,TAG1
           ER       PRINTS
              J         EREXIT     . GO TO ERROR EXIT                *6-11-68WRT
   KEYCHK     LXI,016   14,0  .  ZERO OUT INCREMENT                  *8-05-68WRT
              SZ        13  . ZERO OUT ADJACENT REGISTER             *8-05-68WRT
              DI        13,(6).   . SEE IF KEY ARGUM. ARE MULT OF 6 *6-11-68WRT
              TNZ       14.       . IF NO REMAINDER - O.K.      F 6 *6-11-68WRT
              J         RESTOR.   . RESTORE REG'S. AND BUILD KEYS   *6-11-68WRT
           LA       12,TAG2
           ER       PRINTS
   EREXIT     LX        11,ERRCHK+5. RESTORE REG. 11 FOR W.B.        *6-11-68WRT
              LR        67,('SOPEN ').                               *6-11-68WRT
              SLJ       NERRS     . GO TO WALK - BACK ROUTINE        *6-11-68WRT
   ARGCT      +5        .  NUMBER OF ARGUEMENTS IN CALL              *7-31-68WRT
   RESTOR     LX        11,ERRCHK  . RESTORE STARTING ADD. OF KEYS  *7-31-68WRT
              LX        12,ERRCHK+1 .                                *6-11-68WRT
              DL        13,ERRCHK+2                                  *6-11-68WRT
              LX        15,ERRCHK+4  .                                       40
  .                                                                          41
  . SET THE TABLE ENTRIES FOR THE KEY FIELD DEFINITIONS - WORD POSITION,     42
  . BIT POSITION AND NUMBER OF BITS                                          43
  .
  KEYS        LA        12,0,*11 .  WORD POSITION - SUBTRACT 1 FOR  *8-07-68WRT
             AA         12,(07677777).      TRUE POSITION THEN ADD THE       45
             SA,2       12,1,13.            SORT'S ID CODE 077               46
              LA        12,0,*11 .        BIT POSITION              *8-07-68WRT
             SA,012     12,0,13.                                             48
              LA        12,0,*11 .        NUMBER OF BITS            *8-07-68
             SA,1       12,1,13.                                             50
  .                                                                          51
  . SET THE TABLE ENTRY FOR THE KEY FIELD DEFINITIONS - TYPE                 52
  . (THE 03 BEFORE THE TYPE IS A SORT ID CODE.  THE 00 AFTER THE TYPE IS     53
  . FOR ASCENDING ORDER)                                                     54
```

LINE #   TEXT...

```
 .
          LA        12,0,*11 .    TYPE OF KEY FIELD                 *8-07-68WRT   55
          LA,14     14,030600.          IF TYPE IS 0, CHANGE TO 06 (A)          57
          JZ        12,KEYS1.           FOR ALPHANUMERIC                        58
          LA,14     14,030700.                                                  59
          TNE,14    12,1.               IF TYPE IS 1, CHANGE TO 07 (B)          60
          J         KEYS1                 FOR BINARY 1108 FORMAT                61
          LA,14     14,031100.                                                  62
          TNE,14    12,2.               IF TYPE IS 2, CHANGE TO 011 (D)         63
          J         KEYS1.                FOR SIGNED DECIMAL FORMAT             64
          LA,14     14,032200.                                                  65
          TNE,14    12,3.               IF TYPE IS 3, CHANGE TO 022 (M)         66
          J         KEYS1.                FOR IBM SIGNED DECIMAL FORMAT         67
          LA,14     14,033200.          IF TYPE IS 4, CHANGE TO 032 (U)         68
 KEYS1    SA,2      14,0,13   . FOR UNSIGNED BINARY FORMAT      *7-15-68WRT
 .
 . SET THE TABLE ENTRY FOR THE KEY FIELD DEFINITIONS - ORDER AND                70
 . FIELD NUMBER.                                                                71
 .                                                                              72
          LA,14     12,011.             IF THE ORDER IS 1, CHANGE FROM          73
          TZ        0,*11 .    ASCENDING (SET ABOVE IN TYPE)                    74
          SA,013    12,0,13 .                                   *8-07-68WRT
          LA        12,0,*11 .
          SA,S      12,0,*13       FIELD NUMBER                 *7-9-68WRT
          NOP       0,0,*13 . INCREMENT REGISTER                *7-12-68WRT
 .
          J         KEYS2.              PRINT A 87 ERROR MESSAGE AND            78.4
 MESS87   'MORE THAN 40 KEYS' .                                                 78.9
 .                                                              *7-31-68WRT
 . PROCESS ALL KEY FIELDS UNTIL THE SENTINEL 99999 IS DETECTED.  SET            79
 . END OF TABLE ENTRY THEN RELEASE THE TABLE TO THE SORT.                       80
 .                                                                              81
 KEYS2    LA        12,(99999).         TEST FOR THE SENTINEL                   82
          TE        12,0,11 . IF NOT 99999, GO TO PROCESS THE                   83
          J         KEYS.               NEXT KEY FIELD          *8-07-68WRT
          LA        12,SENTNL.                                                  85
          SA        12,0,13.            END OF TABLE SENTINEL                   86
          LA,14     15,TABLE.           RELEASE TABLE TO THE SORT -             87
          LMJ       11,ROPNS.             RETURN IS TO FPOC ADDRESS             88
 COUNT    +         0                   COUNTER FOR NUMBER OF KEYS              89
 . ***************************************************************************  89.1
 .                                                                              90
 .                                                                              91
 . ROUTINE TO RELEASE A RECORD TO THE SORT                                      92
 .      CALL SRREL (RECORD-ADDRESS, RECORD-LENGTH)                              93
 .                                                                              94
 SRREL*   SX        11,SAVE.            SAVE RETURN ADDRESS (X11)               95
          LA        12,0,11.            RECORD ADDRESS                          96
          LXI       12,*1,11.           LENGTH, IN WORDS, OF RECORD             97
          LMJ       11,RRELS.           RELEASE RECORD TO SORT                  98
                                                                                99
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
                                                                         100
         LX        11,SAVE.              RESTORE X11                     101
         J         3,11.                 RETURN TO USER
 . ............................................................................102
 .                                                                       103
 .                                                                       104
 . ROUTINE TO INFORM THE SORT THAT NO MORE RECORDS ARE TO BE RELEASED.   105
 . THE SORT IS PERFORMED AND CONTROL IS RETURNED TO LPOC.                106
 .     CALL SSORT                                                        107
 .                                                                       108
SSORT.   LMJ       11,RSORTS.            GO TO SORT                      109
 . ............................................................................110
 .                                                                       111
 .                                                                       112
 . ROUTINE TO REQUEST A SORTED OUTPUT RECORD                             113
 .     CALL SRRET(RECORD-ADDRESS, RECORD-LENGTH-ADDRESS, AT-END-ADDRESS) 114
 .                                                                       115
SRRET.   SX        11,SAVE.              SAVE RETURN ADDRESS (X11)       116
         TZ,U2     3,11 . TEST UPPER HALF FOR W.B. WORD        *6-11-68WRT
         TZ,U15    3,11 .  TEST FOR FUNCTION CODE OF 00        *6-11-68WRT
         J         ERROR    . CALL DID NOT HAVE EXACTLY 3 ARG *6-11-68WRT
         LA        12,2,11.              AT-END ADDRESS                  117
         SA        12,$+2.                                               118
         LMJ       11,RRETS.             REQUEST RECORD ADDRESS          119
         +         0.                    SORT RETURN IF NO MORE RECORDS  120
         LX        11,SAVE.              RESTORE X11                     121
         SA        12,SAVE.              RECORD ADDRESS AND LENGTH ARE   122
         LXI,14    12,1.                   RETURNED TO REGISTER 12       123
         LR,2      65,SAVE.                                              124
         SR        65,*1,11              STORE RCD LENGTH IN USER'S FIELD 125
         LX        13,0,11.                                              126
         LXI,14    13,1.                 TRANSFER THE SORTED RECORD TO THE 127
         BT        13,0,*12.               ADDRESS SPECIFIED BU THE USER  128
         J         4,11.                 RETURN TO THE USER              129*
SAVE     +         0.                    TEMP STORAGE FOR SRREL AND SRRET 130
ERROR  LA        12,TAG3
       ER        PRINTS
         LX        11,SAVE . RESTORE LINKAGE ADDRESS FOR W.B.  *6-11-68WRT
         LR        67,(' SRRET') . LOAD NAME OF ROUTINE        *6-11-68
         SLJ       NERRS.  GO TO WALK - BACK ROUTINE           *6-11-68WST
         +2        . NUMBER OF ARGUMENTS WHICH CAUSED ERROR    *6-11-68WRT
 . ............................................................................131
         END.                                                            132
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE LISORT(NVALUE,ARRAY,LOC)
C  SORT POINTER ARRAY BASED ON INTEGER CONTENTS OF ARRAY
C  SORT SEQUENCE IS ASCENDING
      INTEGER ARRAY
      DIMENSION ARRAY(1), LOC(1)
      IBASE = LOC(1) - 1
      DO 100 K = 2, NVALUE
      I = K - 1
      J = K
      J1 = LOC(J) - IBASE
50    I1 = LOC(I) - IBASE
      IF(ARRAY(J1).GE.ARRAY(I1)) GO TO 100
      IHOLD = LOC(J)
      LOC(J) = LOC(I)
      LOC(I) = IHOLD
      J = I
      I = I - 1
      IF(I.GT.0) GO TO 50
100   CONTINUE
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE LSORT(NVALUE,ARRAY,LOC)
      DIMENSION ARRAY(1), LOC(1)
      IBASE = LOC(1) - 1
      DO 100 K = 2, NVALUE
      I = K - 1
      J = K
      J1 = LOC(J) - IBASE
 50   I1 = LOC(I) - IBASE
      IF(ARRAY(J1).LE.ARRAY(I1)) GO TO 100
      IHOLD = LOC(J)
      LOC(J) = LOC(I)
      LOC(I) = IHOLD
      J = I
      I = I - 1
      IF(I.GT.0) GO TO 50
 100  CONTINUE
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      COMPILER (FLD=ABS,R)
      SUBROUTINE ZERMSK(NCASE,MASKS,LEN)
      DIMENSION MASKS(1)
      LEN = (NCASE+34)/35
      DO 5 MSKWRD = 1, LEN
      MASKS(MSKWRD) = 0
5     CONTINUE
      RETURN
      ENTRY SETMSK(ICASE,MASKS)
      LOC = (ICASE+34)/35
      IBIT = MOD(ICASE,35)
      IF(IBIT.EQ.0) IBIT = 35
      IBIT = IABS(IBIT-1)
      FLD(IBIT,1,MASKS(LOC)) = 1
      RETURN
      ENTRY ERSMSK(ICASE,MASKS)
      LOC = (ICASE+34)/35
      IBIT = MOD(ICASE,35)
      IF(IBIT.EQ.0) IBIT = 35
      IBIT = IABS(IBIT-1)
      FLD(IBIT,1,MASKS(LOC)) = 0
      RETURN
      ENTRY CHKMSK(JCASE,MASKS,$)
      LOC = (JCASE+34) / 35
      IBIT = MOD(JCASE,35)
      IF(IBIT.EQ.0) IBIT = 35
      IBIT = IABS(IBIT-1)
      IVAL = FLD(IBIT,1,MASKS(LOC))
      IF(IVAL.EQ.1) RETURN
      RETURN 3
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      COMPILER (FLD=ABS,R)
      SUBROUTINE USEMSK(JCASE,$)
      DIMENSION JA(50), MASKA(1750), ITX(8)
      LOC = (JCASE+34) / 35
      IBIT = MOD(JCASE,35)
      IF(IBIT.EQ.0) IBIT = 35
      IBIT = IABS(IBIT-1)
      IVAL = FLD(IBIT,1,MASKA(LOC))
      IF(IVAL.EQ.1) RETURN
      RETURN 2
      ENTRY GETMSK(ISEQ,ID1,ID2,MATRIX,$)
      IGO = 1
   43 CONTINUE
      REWIND 27
    1 READ(27,END=99) IDREPT, N, (MASKA(I),I=1,N), N, (MASKA(I),
     1 I=1,N), JSEQ, NC, (JA(I),I=1,NC)
      IF(JSEQ.LT.ISEQ) GO TO 1
      GO TO (75,80), IGO
   80 CONTINUE
      MEMB = JA(17)
      ENCODE(123,ITX)
  123 FORMAT(8(' '))
      IDX = JA(36)
      IF(JA(35).EQ.'SAMSEL') IDX = 1
      IF(JA(37).NE.0.AND.IDX.NE.0) ENCODE(124,ITX) (JA(I),I=37,44)
  124 FORMAT(8A6)
   75 IF(JA(1).EQ.ID1.AND.JA(2).EQ.ID2.AND.JA(6).EQ.MATRIX) GO TO 2
      WRITE(6,800) ID1,ID2,MATRIX,   JA(1),JA(2),JA(6)
  800 FORMAT('1FILES DO NOT MATCH!',/,' INPUT FILE = ',
     1 A6,A2,2X,A6,/,' JOB DESCPN = ',A6,A2,2X,A6)
    9 WRITE(6,801)
  801 FORMAT(' THIS REQUEST IS BYPASSED')
      RETURN 5
   99 WRITE(6,802) JSEQ, ISEQ
  802 FORMAT('1END OF FILE ENCOUNTERED ON JOB DESCRIPTION FILE',
     1 ' AFTER REPORT ',I4,' WHILE SEARCHING FOR REPORT ',I4)
      GO TO 9
    2 IF(JSEQ.EQ.ISEQ) RETURN
      WRITE(6,804) JSEQ, ISEQ
  804 FORMAT(' ENCOUNTERED REPORT ',I4,' WHILE SEARCHING FOR',
     1 ' REPORT ',I4,'.  ASSUMING REQUESTED IS REPORT MISSING')
      GO TO 9
      ENTRY GETMSM(ISEQ,ID1,ID2,MATRIX,$,MEMB,IDREPT,ITX)
      IGO = 2
      GO TO 43
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE NEXREL(NAME,IUNIT)
      DIMENSION NAME(3),IMAGE(14)
      IF(NAME(3).NE.' ') ENCODE(300,IHOLD)  NAME(3)
      IF(NAME(3).EQ.' ') ENCODE(301,IHOLD)
300   FORMAT('(-',A1,'). ')
301   FORMAT('.      ')
1     CALL FREES(IUNIT)
      ENCODE(100,IMAGE) (NAME(I),I=1,2), IHOLD
100   FORMAT('@ASG,AZ ',A6,A4,A6,' . ')
      CALL GETFIL(IMAGE)
      ENCODE(101,IMAGE) IUNIT,(NAME(I),I=1,2),IHOLD
101   FORMAT('@USE  ',I2,',',A6,A4,A6,' . ')
      CALL ERTRAN(6,IMAGE)
      RETURN
      ENTRY NEXREM(NAME,IUNIT)
      IF(NAME(3).NE.' ') ENCODE(400,IHOLD)  NAME(3)
      IF(NAME(3).EQ.' ') ENCODE(401,IHOLD)
400   FORMAT('M(-',A1,').')
401   FORMAT('M.    ')
      GO TO 1
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
$(0)
        AXR$ .
        ERU$ .
SAVE    RES      1
BUF              'H,N.M,66,5,0.'
BUF2             'H,N.M,66,6,3.' .
BUF3             'H,N.M,66,0,0.' .
BUF4             'H,N.M,66,6,6.' .
$(1)
NOHEAD.
        SA       A0,SAVE .
        LA       A0,(3,BUF)
        ER       PRTCN$ .
        LA       A0,SAVE .
        J        1,X11 .
MARGIN.
        SA       A0,SAVE .
        LA       A0,(2,BUF2) .
        ER       PRTCN$ .
        LA       A0,SAVE .
        J        1,X11 .
MARGNC.
        SA       A0,SAVE .
        LA       A0,(2,BUF4) .
        ER       PRTCN$ .
        LA       A0,SAVE .
        J        1,X11 .
NOPAGE.
        SA       A0,SAVE .
        LA       A0,(2,BUF3) .
        ER       PRTCN$ .
        LA       A0,SAVE .
        J        1,X11 .
        END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
$(1)      RES     0                      .
          AXR$                           .
NRAND.    LA      A1,*0,X11              . X(I) --> A1.
          MI      A1,MULT                . MULT * X(I) --> (A1,A2).
          DA      A1,INC                 . MULT * X(I) + INC --> (A1,A2).
          AND     A2,(0377777777777)     . (MULT * X(I) + INC) MOD 2**35 --> A3.
          SA      A3,*0,X11              . X(I+1) --> 1ST PARAM OF CALLING SEQ.
          LA,U    A0,155                 .
          LCF     A0,A3                  . FLOAT RESULT --> A1.
          FD      A1,(0244400000000)     . REDUCE TO INTERVAL (0,1)
          SA      A1,*1,X11              . STORE RESULT IN 2ND PARAMETER
          J       3,X11                  . RETURN
.
.         DATA AREA
.
$(0)      RES     0                      .
INC       +       0                      .
INCA      +       7261067085             .
MULT      +       30517578125            .
          END                            .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
$(0)       AXR$                             .
COMPKT     RES       1                      . S1 = ERROR CODE
                                            . S2 = CONSOLE CLASS
                                            . S3 = 0 (FIELDATA)
                                            . H2 = ACTUAL INPUT CHAR COUNT
                                            . S1 = 0
           RES       1                      . S2-S3 = OUTPUT CHAR COUNT
                                            . H2 = OUTPUT BUFFER ADDRESS
                                            . H1 = EXPECTED INPUT CHAR COUNT
           RES       1                      . H2 = INPUT BUFFER ADDRESS
                                            . 0 = OMSG, NONZERO = OMSGW
EXIT       RES       1                      . REG SAVE AREA
SVAO       RES       2                      .
$(1)       ERU$                             .
/.         'OMSG' AND 'OMSGW' SUBROUTINES
 .
 .
OMSG*      .         CALL 'OMSG' USING LEN,MSG,WB.                COBOL
           .         CALL OMSG(LEN,MSG)                           FORTRAN
           .         LMJ       X11,OMSG
           .         +         LEN                 0
           .         +         MSG                 1
           .         +         WB                  2
           .                   (RETURN)            3
           .
           SZ        EXIT
           J         OP1
           .
           .
           .
           .
OMSGW*     .         CALL 'OMSGW' USING LENS,MSGS,LENR,MSGR,WB.   COBOL
           .
           .         CALL OMSGW(LENS,MSGS,LENR,MSGR)
           .
           .         LMJ       X11,OMSGW
           .         +         LENS                0
           .         +         MSGS                1
           .         +         LENR                2
           .         +         MSGR                3
           .         +         WB                  4
           .                   (RETURN)            5
           .
           SX        X11,EXIT
OP1        DS        A0,SVAO
           LA        A0,(000100,0)      . FORCE MESSAGE TO "LOAD" CONSOLE
           SA        A0,COMPKT          . STORE VALUE IN COMMUNICATION PACKET
           SZ        COMPKT+2           .
           LA        A0,*0,X11          . A0 = LENGTH OF MESSAGE TO BE SENT
           MSI,U     A0,6               .
           SA,H1     A0,COMPKT+1        . SET OUTPUT CHAR COUNT
           LA        A0,1,X11           .
           SA,H2     A0,COMPKT+1        . SET OUTPUT BUFF ADDRESS
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
            TNZ      EXIT              . DO WE HAVE TO SET INPUT?
            J        OP2               . NO
            LA       A0,*2,X11         . YES, A0 = EXPECTED LENGTH
            MSI,U    A0,6              .
            SA,H1    A0,COMPKT+2       . SET EXPECTED INPUT CHAR COUNT
            LA       A0,3,X11          .
            SA,H2    A0,COMPKT+2       . SET INPUT BUFF ADDRESS
   OP2      LA,U     A0,COMPKT         .
            ER       COM$              . SEND MESSAGE AND IF OMSGW WAIT FOR REPLY
            TZ       EXIT              . WERE WE EXPECTING A RESPONSE?
            J        OP3               . YES, GO SET ACTUAL CHAR COUNT
            LA       A0,SVA0           .
            J        3,X11             . EXIT FROM OMSG
   OP3      LA,H2    A0,COMPKT         . A0 = ACTUAL CHARACTER COUNT
            AA,U     A0,5              .
            DSL      A0,36             .
            DI,U     A0,6              .
            SA       A0,*2,X11         . SET LENR TO ACTUAL LENGTH OF RESPONSE
            DL       A0,SVA0           .
            J        5,X11             . EXIT FROM OMSGW
            END                        .
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
$(0)      AXR$                        .
        . THIS IS A MODIFIED VERSION OF DICK ROGERS OMSG ROUTINE
        . (WHICH WOULD DISPLAY MESSAGES ON THE "LOAD" MESSAGE
        . CONSOLE.)  THIS MODIFICATION WILL PUT THE MESSAGES BACK
        . ONTO CONSOLE 1 AS DID THE ORIGINAL VERSION.  THIS MOD-
        . IFICATION WILL BE USED WITH INTERACTIVE PROCESSORS TO
        . INSURE THAT NECESSARY OPERATOR LOAD MESSAGES ARE NOT
        . ROLLED OFF THE CONSOLE 0 SCREEN TOO QUICKLY.  THE MAIN
        . CHANGE IS THAT THE ENTRY POINTS ARE FOLLOWED BY A "O",
        . GIVING "OMSGO" AND "OMSGWO".
COMPKT    RES       1                 . S1 = ERROR CODE
                                      . S2 = CONSOLE CLASS
                                      . S3 = 0 (FIELDATA)
                                      . H2 = ACTUAL INPUT CHAR COUNT
          RES       1                 . S1 = 0
                                      . S2-S3 = OUTPUT CHAR COUNT
                                      . H2 = OUTPUT BUFFER ADDRESS
          RES       1                 . H1 = EXPECTED INPUT CHAR COUNT
                                      . H2 = INPUT BUFFER ADDRESS
EXIT      RES       1                 . 0 = OMSG, NONZERO = OMSGWO
SVAO      RES       2                 . REG SAVE AREA
$(1)      ERU$                        .
/.        'OMSG' AND 'OMSGW' SUBROUTINES
 .
 .
OMSGO.    .         CALL 'OMSGO' USING LEN,MSG,WB.                     COBOL
          .         CALL OMSGO(LEN,MSG)                                 FORTRAN
          .         LMJ       X11,OMSGO
          .         +         LEN                0
          .         +         MSG                1
          .         +         WB                 2
          .                   (RETURN)           3
          SZ        EXIT
          J         OP1
          .
          .
          .
          .
OMSGWO.   .         CALL 'OMSGWO' USING LENS,MSGS,LENR,MSGR,WB.        COBOL
          .
          .         CALL OMSGWO(LENS,MSGS,LENR,MSGR)
          .
          .         LMJ       X11,OMSGWO
          .         +         LENS               0
          .         +         MSGS               1
          .         +         LENR               2
          .         +         MSGR               3
          .         +         WB                 4
          .                   (RETURN)           5
          SX        X11,EXIT
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
OP1     DS       AO,SVAO
        LA       AO,(0,0)         . FORCE MESSAGE TO CONSOLE #1
        SA       AO,COMPKT        . STORE VALUE IN COMMUNICATION PACKET
        SZ       COMPKT+2         .
        LA       AO,*0,X11        . AO = LENGTH OF MESSAGE TO BE SENT
        MSI,U    AO,6             .
        SA,H1    AO,COMPKT+1      . SET OUTPUT CHAR COUNT
        LA       AO,1,X11         .
        SA,H2    AO,COMPKT+1      . SET OUTPUT BUFF ADDRESS
        TNZ      EXIT             . DO WE HAVE TO SET INPUT?
        J        OP2              . NO
        LA       AO,*2,X11        . YES, AO = EXPECTED LENGTH
        MSI,U    AO,6             .
        SA,H1    AO,COMPKT+2      . SET EXPECTED INPUT CHAR COUNT
        LA       AO,3,X11         .
        SA,H2    AO,COMPKT+2      . SET INPUT BUFF ADDRESS
OP2     LA,U     AO,COMPKT        .
        ER       COM$             . SEND MESSAGE AND IF OMSGWO WAIT FOR REPL
        TZ       EXIT             . WERE WE EXPECTING A RESPONSE?
        J        OP3              . YES, GO SET ACTUAL CHAR COUNT
        LA       AO,SVAO          .
        J        3,X11            . EXIT FROM OMSGO
OP3     LA,H2    AO,COMPKT        . AO = ACTUAL CHARACTER COUNT
        AA,U     AO,5             .
        DSL      AO,36            .
        DI,U     AO,6             .
        SA       AO,*2,X11        . SET LENR TO ACTUAL LENGTH OF RESPONSE
        DL       AO,SVAO          .
        J        5,X11            . EXIT FROM OMSGWO
        END                       .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
C     THIS ROUTINE READS THE VARIABLE DICTIONARY FROM THE UNIT
C     SPECIFIED BY 'IUNIT'.  IT WRITES THE DICTIONARY ON THE UNIT
C     SPECIFIED BY 'IOUT'.  THIS ROUTINE EXPECTS THE CALLING PROGRAM
C     TO POSITION/OPEN/CLOSE ALL TAPES AS REQUIRED.
      SUBROUTINE PRTDIC(IUNIT,IOUT,NTITLE,ILOC,NDICT)
      DIMENSION ILOC(1),DICT(15,20)
      NWORDS=NTITLE*14
      IPAGE=1
      LCNT=NTITLE+7
      WRITE(IOUT,10) IPAGE
   10 FORMAT('1VARIABLE DICTIONARY',58X,'DICT  ',' PAGE ',I3,//)
      IF(NTITLE.EQ.0) GO TO 100
      WRITE(IOUT,20) (ILOC(I),I=1,NWORDS)
   20 FORMAT(X,13A6,A2)
  100 WRITE(IOUT,30)
   30 FORMAT(//,' IDENT',20X,'VARIABLE DEFINITION',40X,'FORMAT',/)
      DO 120 IX=1,NDICT
      READ(IUNIT) L,((DICT(I,K),I=1,15),K=1,L)
      DO 110 IK=1,L
      WRITE(IOUT,40)(DICT(K,IK),K=1,15)
   40 FORMAT(X,15A6)
      LCNT=LCNT+1
      IF(LCNT.NE.55) GO TO 110
      IPAGE=IPAGE+1
      WRITE(IOUT,10) IPAGE
      WRITE(IOUT,30)
      LCNT=7
  110 CONTINUE
  120 CONTINUE
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE IRAND(JSEED,NPOP,NSAMP)
      POP = NPOP
      SAMP = NSAMP
      ISEED = JSEED
      CHOSEN =  0.0
      VIEWED =  0.0
      RETURN
      ENTRY SRAND(INSAMP)
      INSAMP = -1
      VIEWED = VIEWED + 1.0
      SPROB = (SAMP-CHOSEN)/(POP-VIEWED+1.0)
      CALL NRAND(ISEED,TPROB)
      IF(TPROB.LT.SPROB) GO TO 50
      RETURN
 50   INSAMP = 1
      CHOSEN = CHOSEN + 1.0
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      FUNCTION RANF(X)
      CALL GETRAN(ANS)
      RANF = ANS
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
      SUBROUTINE RANSEQ(LENGTH,LIST)
      DIMENSION LIST(1),IPOINT(300)
      DATA IFIRST/1/
      IF(IFIRST.EQ.0) GO TO 5
      CALL ERTRAN(9,IDATE,ITIME)
      MATCH = AND(IDATE,ITIME)
      ISEED = MATCH + ITIME + 31
      IFIRST = 0
    5 CONTINUE
      IF(LENGTH.GT.300) GO TO 900
      DO 6 I = 1, LENGTH
      CALL NRAND(ISEED,R)
      IPOINT(I) = R*10000.
      FLD(0,18,IPOINT(I)) = I
    6 CONTINUE
      CALL H2SORT(LENGTH,IPOINT)
      DO 7 I = 1, LENGTH
      INDEX = FLD(0,18,IPOINT(I))
      IPOINT(I) = LIST(INDEX)
    7 CONTINUE
      DO 8 I = 1, LENGTH
      LIST(I) = IPOINT(I)
    8 CONTINUE
      RETURN
  900 CONTINUE
      WRITE(6,101) LENGTH
  101 FORMAT('1RANDOM SORT IS NOT ALLOWED ON MORE THAN 300 CASES.',
     1 /,' REQUEST WAS MADE TO SORT ',I6,'.  RUN ABORTS.')
      CALL ERTRAN(2)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

REPEND CLOSE REPORTS AND PUNCH A REQUEST CARD

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE REPEND(ISTUDY,IDREPT,LABEL,IDUM)
      DIMENSION LABEL(8)
      REWIND 2
      READ(2) JSEQ
      WRITE(1,100) ISTUDY, IDREPT, LABEL, JSEQ
  100 FORMAT(A4,A6,8A6,T76,I5)
      JSEQ = JSEQ + 1
      IDUM = JSEQ
      WRITE(26,101) JSEQ
  101 FORMAT('BEGIN REPORT',A6)
      REWIND 2
      WRITE(2) JSEQ
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE RESDRV(IUNIT)
      DIMENSION IMAGE(7), IFMAT(6)
      DATA IMAGE/'RESERVE TAPE DRIVE, ENTER NUMBER (0-9)'/
      DATA IBLANK/' '/
      DATA IFMAT/'(A1)','(1XA1)(2XA1)(3XA1)(4XA1)(5XA1)'/
      LEN = 1
      CALL OMSGW(7,IMAGE,LEN,JUNIT)
      DO 10 I = 1, 6
      IMAT = IFMAT(I)
      DECODE(IMAT,JUNIT) IUNIT
      IF(IUNIT.EQ.IBLANK) GO TO 10
      CALL BCDBIN(IUNIT,IUNIT,1,$10)
      RETURN
10    CONTINUE
      WRITE(6,100) JUNIT
      STOP
100   FORMAT(' OPERATOR DID NOT RESERVE A VALID UNIT (',A6,')')
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE ERRMSG(M,JIST)
      DIMENSION LIST(14), JIST(9)
      LINE = LINE + 1
      IF(LINE.LE.60) GO TO 1
      ISW = 1
      GO TO 3
 1    NN = M
      IF(NN.GT.9) NN = 9
      WRITE(6,100) (JIST(I),I=1,NN)
      RETURN
      ENTRY PRINTC
      LINE = LINE + 2
      IF(LINE.LE.59) GO TO 2
      ISW = 2
      GO TO 3
 2    READ(0,101) LIST
      WRITE(6,102) LIST
      RETURN
      ENTRY RESET
      IPAGE = 0
      ISW = 3
 3    LINE = 5
      IPAGE = IPAGE + 1
      WRITE(6,103) IPAGE
      GO TO (1,2,4), ISW
 4    RETURN
 100  FORMAT(1X,9A6)
 101  FORMAT(13A6,A2)
 102  FORMAT('0   *',13A6,A2,'*')
 103  FORMAT('1PROGRAM CONTROL CARD LISTING',21X,'PAGE',I5,/)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

TEXT...

```
      SUBROUTINE RPCOPY(JSEQ)
      DIMENSION LOC(22)
      CALL REWIND
      REWIND 26
1     CALL TREAD(NWORD,LOC,JSEQ,$2,$3)
2     WRITE(26,100) (LOC(I),I=1,NWORD)
      GO TO 1
3     WRITE(2) JSEQ
      RETURN
100   FORMAT(22A6)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

RPEOF LOCATES AN END-OF-FILE ON A REPORT FILE

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE RPEOF(IUNIT,JSEQ)
      DIMENSION ID(3)
      EQUIVALENCE (ID(3),LSEQ)
      REWIND 2
      READ(2) MSEQ
      REWIND IUNIT
1     READ(IUNIT,100,END=2) ID
100   FORMAT(3A6)
      IF(ID(1).NE.'BEGIN ') GO TO 1
      IF(ID(2).NE.'REPORT') GO TO 1
      IF(LSEQ.NE.MSEQ) GO TO 1
      JSEQ = LSEQ
      WRITE(6,102) JSEQ
102   FORMAT(' LAST REPORT SENTINEL ON FILE WAS ',I6)
      RETURN
2     WRITE(6,110) MSEQ, LSEQ
110   FORMAT(' UNEXPECTED END-OF-FILE ON REPORT TAPE WHILE SEARCH',
     1 'ING FOR REPORT ',I6,/,' THIRD WORD OF LAST RECORD = ',A6)
      CALL ERTRAN(2)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      SUBROUTINE RPINDX
      DIMENSION LOC(23),  NAME(3)
      INTEGER PAGE, REPORT, ONE,  TEMP
      COMMON/INFILE/ IDPROG, NAME
      DATA IBLANK, ONE /' ','1'/
      PAGE = 0
      REPORT = 0
    1 CALL THEAD(NWORD,LOC,JSEQ,$2,$777)
      GO TO 1
    2 CONTINUE
      REPORT = REPORT + 1
   31 CALL THEAD(NEWORD,LOC,JSEQ,$666,$4)
      DECODE(101,LOC) TEMP
      ENCODE(101,LOC) TEMP
      IF(MOD(REPORT,52).NE.1) GO TO 3
      PAGE = PAGE + 1
      WRITE(6,115) NAME, PAGE
    3 WRITE(6,104) LOC(14), (LOC(I),I=1,9), JSEQ
      GO TO 1
    4 WRITE(6,105)
      RETURN
  666 CONTINUE
      LSEQ = JSEQ - 1
      IF(MOD(REPORT,52).NE.1) GO TO 32
      PAGE = PAGE + 1
      WRITE(6,115) NAME, PAGE
   32 WRITE(6,667) LSEQ
  667 FORMAT(32X,'*******',4X,'NO REPORT',40X,I5)
      GO TO 31
  777 WRITE(6,106)
      RETURN
  101 FORMAT(1X,A5)
  104 FORMAT(32X,A6,3X,8A6,1A3,I5)
  105 FORMAT(/,,32X,'END OF REPORT FILE')
  106 FORMAT('0LAST REPORT SENTINEL IS MISSING.')
  115 FORMAT('1',47X,'INDEX OF REPORT FILE ',3A6,10X,'PAGE',I3,//,29X,
     1'REPORT ID     REPORT TITLE',37X,'REPORT',/,91X,'NUMBER',/)
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE SAMPLE(IDS,IDVAR,NVAR,IERR,IPNT)
      DIMENSION LCHAR(78), IDVAR(9), LSEQ(4,8), LPRES(4), LVALS(1)
      DATA JSW, LPRES/0,')',':','&','('/
      IF(JSW.NE.0) GO TO 13
      ENCODE(100,LCHAR)
100   FORMAT('@ASG,T 4.,F/0/TRK/50 . ')
      CALL GETMAS(LCHAR)
      ITP = 4
13    JSW = 1
      NOPS = 0
      IVAR = 0
      IPRS = 0
      READ(5,101,END=900) NV, LCHAR
      IF(IPNT.EQ.1) CALL PRINTC
101   FORMAT(A2,78A1)
      IF(NV.EQ.'EN') GO TO 600
      INDEX = 0
      ISW = 0
1     INDEX = INDEX + 1
      IF(INDEX.GT.78) GO TO 15
      ICHAR = LCHAR(INDEX)
2     IF(ICHAR.EQ.' ') GO TO 1
      IF(ICHAR.EQ.'V'.OR.ICHAR.EQ.'C') GO TO 10
      DO 3 ITEST = 1, 4
      IF(ICHAR.EQ.LPRES(ITEST)) GO TO (5,6,6,4), ITEST
3     CONTINUE
900   IERR = IERR + 1
      CALL BCDBIN(NV,NVAR,2,$901)
901   CONTINUE
      RETURN
C  FOUND "(" - INCREMENT PRECEDENCE
4        IPRS = IPRS + 4
         GO TO 1
C  FOUND ")" - REDUCE PRECEDENCE
5        IPRS = IPRS - 4
         IF(IPRS.LT.0) GO TO 900
         GO TO 1
C  FOUND A VALID OPERATOR "&" OR ":"
6        NOPS = NOPS + 1
         ISW = 0
         IF(NOPS.GT.8) GO TO 900
         LSEQ(1,NOPS) = ITEST + IPRS
         LSEQ(2,NOPS) = ITEST
         LSEQ(3,NOPS) = NOPS
         LSEQ(4,NOPS) = NOPS + 1
         GO TO 1
C  FOUND A VARIABLE-ID
10       IF(ISW.NE.0) GO TO 900
         ITYPE = 1
         IF(ICHAR.EQ.'V') ITYPE = -1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          ISUM = 0
          ISW = 1
          IVAR = IVAR + 1
    11    INDEX = INDEX + 1
          IF(INDEX.GT.78) GO TO 12
          ICHAR = LCHAR(INDEX)
          IF(ICHAR.EQ.' ') GO TO 11
          CALL BCDBIN(ICHAR,IVAL,1,$12)
          ISUM = ISUM*10 + IVAL
          GO TO 11
    12    CONTINUE
          IF(ISUM.LE.0.OR.ISUM.GT.999) GO TO 900
          IDVAR(IVAR) = ITYPE*ISUM
          INDEX = INDEX - 1
          GO TO 1
C   END OF INITIAL SCAN
    15    CONTINUE
          IF(IPRS.NE.0) GO TO 900
          NVAR = IVAR
          IF(NOPS.LT.2) GO TO 50
C   SORT PRECEDENCES INTO DESCENDING ORDER
          DO 18 K = 2, NOPS
          I = K -1
          J = K
    16    IF(LSEQ(1,I).GE.LSEQ(1,J)) GO TO 18
          DO 17 L = 1,4
          IHOLD = LSEQ(L,J)
          LSEQ(L,J) = LSEQ(L,I)
          LSEQ(L,I) = IHOLD
    17    CONTINUE
          J = I
          I = I - 1
          IF(I.GT.0) GO TO 16
    18    CONTINUE
C   COLLAPSE OPERAND IDS LIKE GROUP
          DO 20 K = 2, NOPS
          I = K - 1
          ID2 = LSEQ(4,I)
          ID1 = LSEQ(3,I)
          DO 19 L = K, NVAR
          IF(LSEQ(3,L).EQ.ID2) LSEQ(3,L) = ID1
    19    CONTINUE
    20    CONTINUE
    21    WRITE(ITP) IDS,NOPS,((LSEQ(I,J),I=2,4),J=1,NOPS)
          RETURN
    50    IF(NOPS.EQ.1) GO TO 21
          IF(NVAR.EQ.0) GO TO 900
          NOPS = 1
          LSEQ(2,1) = 1
          LSEQ(3,1) = 1
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      GO TO 21
600   CONTINUE
      IF(LCHAR(1).NE.'D') GO TO 900
      NVAR = 0
      NOPS = 1
      LSEQ(2,1) = 4
      IERR = -1
      GO TO 21
      ENTRY SELECT(IDS,LVALS,$,$)
      IF(JSW.NE.1) GO TO 60
      ENDFILE ITP
      JSW = 2
60    REWIND ITP
61    READ(ITP,END=777) IDT, NOPS, ((LSEQ(I,J),I=2,4),J=1,NOPS)
      IF(IDT.NE.IDS) GO TO 61
      REWIND ITP
      DO 65 IOP = 1, NOPS
      IP = LSEQ(2,IOP)
      I1 = LSEQ(3,IOP)
      I2 = LSEQ(4,IOP)
      GO TO (62,63,64,66), IP
C  SELECT ON A SINGLE VARIABLE
62    IF(LVALS(I1).NE.1) RETURN 3
      RETURN
C  PERFORM A LOGICAL "OR"
63    LVALS(I1) = LVALS(I1) + LVALS(I2)
      GO TO 65
C  PERFORM A LOGICAL "AND"
64    LVALS(I1) = LVALS(I1) * LVALS(I2)
65    CONTINUE
      IF(LVALS(1).EQ.0) RETURN 3
C  IMMEDIATE RETURN ON NO VARIABLES
66    RETURN
777   RETURN 4
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE SETUP(I1,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,JSEQ)
      DIMENSION LIST(6),  ID(3,6), IMAGE(7), NAME(3), INPUT(3)
      DIMENSION IDATE(2), ITIME(2)
      INTEGER BLANK, UNIT
      COMMON/INFILE/ IDPROG, INPUT
      DATA BLANK/' '/, ISCRTH/'SCRATC'/
      JSEQ = 0
      LIST(1) = I1
      LIST(2) = I2
      LIST(3) = I3
      LIST(4) = I4
      LIST(5) = I5
      LIST(6) = I6
      READ(5,800) IDPROG, ID
800   FORMAT(A6,1X,6(A6,A4,A1,1X))
      CALL DATETM(IDATE,ITIME)
      WRITE(6,112) IDPROG, IDATE, ITIME
      DO 99 L = 6, 1, -1
      IF(LIST(L).NE.26) GO TO 99
      IF(ID(1,L).EQ.' ') GO TO 99
      IF(ID(1,L).NE.'SCRATC') GO TO 80
      ENCODE(108,IMAGE) LIST(L)
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      ENCODE(106,IMAGE) L
      WRITE(6,111) (IMAGE(I),I=1,4)
      CALL GETFIL(IMAGE)
      ENCODE(105,IMAGE) LIST(L), L
      WRITE(6,111) (IMAGE(I),I=1,4)
      CALL ERTRAN(6,IMAGE)
      GO TO 99
80    CONTINUE
      UNIT = LIST(L)
          LIST(L) = 0
          CALL CYCLES(ID(1,L),UNIT,IASGD)
          IF(IASGD.EQ.1) GO TO 97
90        IOLD = 28
          CALL RPCOPY(JSEQ)
          CALL FREE(IOLD)
          GO TO 99
97        CALL RPEOF(UNIT,JSEQ)
99        CONTINUE
      DO 49 L = 1, 6
      IF(LIST(L).EQ.0) GO TO 49
      IF(ID(1,L).EQ.BLANK.AND.ID(2,L).EQ.BLANK) GO TO 16
      UNIT = LIST(L)
      IF(ID(1,L).EQ.ISCRTH) GO TO 15
      IF(UNIT.GT.28.OR.UNIT.LT.10) GO TO 777
      IF(UNIT.GT.14) GO TO 4
      IF(ID(3,L).EQ.BLANK) GO TO 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

SETUP THE STANDARD FILE SETUP SUBROUTINE

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
          ENCODE(100,NAME)    (ID(I,L),I=1,3)
          GO TO 2
    1     ENCODE(101,NAME)    (ID(I,L),I=1,2)
    2     CONTINUE
          IF(UNIT.EQ.10) INPUT(1) = NAME(1)
          IF(UNIT.EQ.10) INPUT(2) = NAME(2)
          IF(UNIT.EQ.10) INPUT(3) = NAME(3)
          CALL ASGA(ID(1,L),UNIT)
          GO TO 49
    4     IF(UNIT.GT.19) GO TO 6
          CALL ASGC(ID(1,L),UNIT)
          GO TO 49
    6     IF(UNIT.GT.24) GO TO 14
   15     ENCODE(108,IMAGE) UNIT
          WRITE(6,111) (IMAGE(I),I=1,3)
          CALL ERTRAN(6,IMAGE)
          ENCODE(106,IMAGE) L
          WRITE(6,111) (IMAGE(I),I=1,4)
          CALL GETFIL(IMAGE)
          ENCODE(105,IMAGE) UNIT, L
          WRITE(6,111) (IMAGE(I),I=1,4)
                CALL ERTRAN(6,IMAGE)
                GO TO 49
   16     ENCODE(108,IMAGE) LIST(L)
          WRITE(6,111) (IMAGE(I),I=1,3)
          CALL ERTRAN(6,IMAGE)
          ENCODE(115,IMAGE) LIST(L), L
          WRITE(6,111) (IMAGE(I),I=1,3)
          CALL ERTRAN(6,IMAGE)
          GO TO 49
   14     CALL ASGAM(ID(1,L),UNIT)
          IF(ID(3,L).EQ.BLANK) ID(3,L) = '0'
          ENCODE(177,INPUT) (ID(I,L),I=1,3)
  177     FORMAT(A6,A4,'M(-',A1,')',)
   49     CONTINUE
          LIST(1) = J1
          LIST(2) = J2
          LIST(3) = J3
          LIST(4) = J4
          LIST(5) = J5
          DO 50 I = 1, 5
          IF(LIST(I).EQ.0) GO TO 50
          ENCODE(109,IMAGE) LIST(I)
          WRITE(6,111) (IMAGE(KK),KK=1,5)
          CALL GETFIL(IMAGE)
   50     CONTINUE
          CALL NOHEAD
          RETURN
  777     WRITE(6,110)
          CALL ERTRAN(2)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
100   FORMAT( A6,A4,2H(-,A1,4H) . )
101   FORMAT( A6,A4,3H . )
105   FORMAT('@USE ',I2,',SCRATCH-',I1,' . ')
106   FORMAT(16H@ASG,TZ SCRATCH-,I1,7H,16N . )
108   FORMAT(8H@FREE,A ,I2,3H . )
109   FORMAT(8H@ASG,TZ ,I2,',F/O/POS/50    . ')
110   FORMAT('1INVALID UNIT FOR SETUP SUBROUTINE. RUN HALTS.')
111   FORMAT(1X,7A6)
112   FORMAT('1DYNAMICALLY ISSUED CONTROL STATEMENTS FOR ',A6,51X,
     1 A6,A3,1X,A6,A2,//)
115   FORMAT('@USE ',I2,',FILE',I1,' . ')
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

SETUP9 FILE ASSIGNMENT ROUTINE FOR JOBSPC & JOBGRP

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE SETUP9(I1,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,JSEQ)
      DIMENSION LIST(6),  ID(3,6), IMAGE(7), NAME(3), INPUT(3)
      DIMENSION IDATE(2), ITIME(2)
      INTEGER BLANK, UNIT
      COMMON/INFILE/ IDPROG, INPUT
      DATA BLANK/' '/, ISCRTH/'SCRATC'/
      JSEQ = 0
      LIST(1) = I1
      LIST(2) = I2
      LIST(3) = I3
      LIST(4) = I4
      LIST(5) = I5
      LIST(6) = I6
      READ(5,800) IDPROG, ID
  800 FORMAT(A6,1X,6(A6,A4,A1,1X))
      CALL DATETM(IDATE,ITIME)
      WRITE(6,112) IDPROG, IDATE, ITIME
      DO 99 L = 6, 1, -1
      IF(LIST(L).NE.25.AND.LIST(L).NE.26) GO TO 99
      IF(ID(1,L).EQ.' ') GO TO 99
      IF(ID(1,L).NE.'SCRATC') GO TO 80
      ENCODE(108,IMAGE) LIST(L)
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      ENCODE(106,IMAGE) L
      WRITE(6,111) (IMAGE(I),I=1,4)
      CALL GETFIL(IMAGE)
      ENCODE(105,IMAGE) LIST(L), L
      WRITE(6,111) (IMAGE(I),I=1,4)
      CALL ERTRAN(6,IMAGE)
      GO TO 99
   80 CONTINUE
      UNIT = LIST(L)
           LIST(L) = 0
           CALL CYCLES(ID(1,L),UNIT,IASGD)
           IF(IASGD.EQ.1) GO TO 95
           IF(UNIT.NE.25) GO TO 90
           IOLD = 27
           CALL JDCOPY
           CALL FREE(IOLD)
           GO TO 99
   90      IOLD = 28
           CALL RPCOPY(JSEQ)
           CALL FREE(IOLD)
           GO TO 99
   95      IF(UNIT.NE.25) GO TO 97
           CALL JDEOF
           GO TO 99
   97      CALL RPEOF(UNIT,JSEQ)
   99      CONTINUE
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
      DO 49 L = 1, 6
      IF(LIST(L).EQ.0) GO TO 49
      IF(ID(1,L).EQ.BLANK.AND.ID(2,L).EQ.BLANK) GO TO 16
      UNIT = LIST(L)
      IF(ID(1,L).EQ.ISCRTH) GO TO 15
      IF(UNIT.GT.28.OR.UNIT.LT.10) GO TO 777
      IF(UNIT.GT.14) GO TO 4
      IF(ID(3,L).EQ.BLANK) GO TO 1
      ENCODE(100,NAME)   (ID(I,L),I=1,3)
      GO TO 2
 1    ENCODE(101,NAME)   (ID(I,L),I=1,2)
 2    CONTINUE
      IF(UNIT.EQ.10) INPUT(1) = NAME(1)
      IF(UNIT.EQ.10) INPUT(2) = NAME(2)
      IF(UNIT.EQ.10) INPUT(3) = NAME(3)
      CALL ASGA(ID(1,L),UNIT)
      GO TO 49
 4    IF(UNIT.GT.19) GO TO 6
      CALL ASGC(ID(1,L),UNIT)
      GO TO 49
 6    IF(UNIT.GT.24) GO TO 14
 15   ENCODE(108,IMAGE) UNIT
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      ENCODE(106,IMAGE) L
      WRITE(6,111) (IMAGE(I),I=1,4)
      CALL GETFIL(IMAGE)
      ENCODE(105,IMAGE) UNIT, L
      WRITE(6,111) (IMAGE(I),I=1,4)
          CALL ERTRAN(6,IMAGE)
          GO TO 49
 16   ENCODE(108,IMAGE) LIST(L)
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      ENCODE(115,IMAGE) LIST(L), L
      WRITE(6,111) (IMAGE(I),I=1,3)
      CALL ERTRAN(6,IMAGE)
      GO TO 49
 14   CALL ASGAM(ID(1,L),UNIT)
      IF(ID(3,L).EQ.BLANK) ID(3,L) = '0'
      ENCODE(177,INPUT) (ID(I,L),I=1,3)
 177  FORMAT(A6,A4,'M(-',A1,')',)
 49   CONTINUE
      LIST(1) = J1
      LIST(2) = J2
      LIST(3) = J3
      LIST(4) = J4
      LIST(5) = J5
      DO 50 I = 1, 5
      IF(LIST(I).EQ.0) GO TO 50
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
           ENCODE(109,IMAGE) LIST(I)
           WRITE(6,111) (IMAGE(KK),KK=1,5)
           CALL GETFIL(IMAGE)
   50      CONTINUE
           CALL NOHEAD
           RETURN
   777     WRITE(6,110)
           CALL ERTRAN(2)
   100     FORMAT( A6,A4,2H(-,A1,4H) . )
   101     FORMAT( A6,A4,3H . )
   105     FORMAT('@USE ',I2,',SCRATCH-',I1,' . ')
   106     FORMAT(16H@ASG,TZ SCRATCH-,I1,7H,16N . )
   108     FORMAT(8H@FREE,A ,I2,3H . )
   109     FORMAT(8H@ASG,TZ ,I2,',F/O/POS/50    . ')
   110     FORMAT('1INVALID UNIT FOR SETUP SUBROUTINE. RUN HALTS.')
   111     FORMAT(1X,7A6)
   112     FORMAT('1DYNAMICALLY ISSUED CONTROL STATEMENTS FOR ',A6,51X,
          1 A6,A3,1X,A6,A2,//)
   115     FORMAT('@USE ',I2,',FILE',I1,' . ')
           END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE STATS(IVAL)
      DOUBLE PRECISION SUM, SUMSQ
      DATA IFIRST/1/
      GO TO (1,10), IFIRST
 1    CONTINUE
      IFIRST = 2
      KOUNT = 1
      MAXVAL = IVAL
      MINVAL = IVAL
      SUM = IVAL
      SUMSQ = IVAL**2
      RETURN
 10   CONTINUE
      KOUNT = KOUNT + 1
      SUM = SUM + IVAL
      SUMSQ = SUMSQ + IVAL**2
      IF(IVAL.GT.MAXVAL) MAXVAL = IVAL
      IF(IVAL.LT.MINVAL) MINVAL = IVAL
      RETURN
      ENTRY STATSS(AMEAN,SD,MINT,MAXT,ISUM,K)
      X = KOUNT
      K = KOUNT
      ISUM = SUM
      AMEAN = SUM/X
      SD = SQRT(ABS(SUMSQ/X - AMEAN*AMEAN))
      MINT = MINVAL
      MAXT = MAXVAL
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...

 $(0)    AXR$
 PKT     RES      13
 PF      FORM     12,6,18
 SV      RES      3
 $(1)    ERU$
 STATUS*
         DS       A0,SV       .  SAVE A0,A1
         SZ       PKT+8       .  SET REEL COUNT, ETC TO ZERO
         SZ       PKT+6       .  SET EQUIPMENT CODE, ETC TO ZERO
         DL       A0,*0,X11     .  RETREIVE FILENAME
         DS       A0,PKT        .  STORE FILENAME IN PACKET
         LA       A0,(13,PKT)
         ER       FITEM$
         LA,U     A0,014      .   SET A0 = EQUIP CODE OF 16N
         TE,S1    A0,PKT+6    .   IS THE FILE ON A 16N DRIVE?
         J        ERROR       .   IF NOT, JUMP TO INDICATE ERROR
         LA       A0,PKT+11  .     PICK UP CURRENT REEL-ID
         SA       A0,*1,X11     .    RETURN CURRENT REEL-ID
         LA,S1    A0,PKT+8      .    RETREIVE NUMBER OF REELS
         TE,U     A0,2          .    IS THIS A TWO REEL FILE?
         SZ       PKT+12        .    SET REEL-ID OF NEXT REEL TO ZERO
         LA       A0,PKT+12      .     PICK UP NEXT REEL-ID
         SA       A0,*2,X11     .    RETURN NEXT REEL-ID
         LA,S1    A0,PKT+7       .     PICK UP TEMP/CAT FLAG
         SSL      A0,4         .         SHIFT 'DOWN' BIT OUT OF WORD
         LSSL     A0,35        .         GET RID OF LABEL CHECK BIT
         SSL      A0,35       .        RETURN TEMP/CAT FLAG
         SA       A0,*3,X11  .         GIVE THE FLAG TO THE MAIN PROGRAM
         DL       A0,SV       .  RESTORE REGISTERS
         J        5,X11       .      RETURN
 ERROR   SZ       *1,X11        .    SET REEL-ID1 TO ZERO
         SZ       *2,X11        .    SET REEL-ID2 TO ZERO
         LA,U     A0,1          .    SET A0 = 1
         SA       A0,*3,X11     .    SET TEMPORARY FLAG = 1 (NOT ASSIGNED)
         DL       A0,SV         .    RESTORE REGISTERS
         J        5,X11          .    RETURN TO USER
         END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
 $(0)      AXR$
 ARG       FORM      12,6,18              . STORAGE AREA FOR ALL SUBROUTINES
 P         PROC                           .
 SAVEA*    NAME      0                    . SAVE/RESTORE REGISTER PROCEDURE
 SAVEX*    NAME      16                   . A REGISTERS IN SV+0 THRU SV+15
 SAVER*    NAME      27                   . X REGISTERS IN SV+16 THRU SV+26
 K        *DO P(1) , S  P(1,K),SV+K-1+P(0,0)+P(0,1)*41 .
           GO        XIT                  . EXIT FROM PROC
 RESTA*    NAME      0                    . A REGISTERS IN SV+0 THRU SV+15
 RESTX*    NAME      16                   . X REGISTERS IN SV+16 THRU SV+26
 RESTR*    NAME      27                   . R REGISTERS IN SV+27 THRU SV+40
 K         DO P(1) , L  P(1,K),SV+K-1+P(0,0)+P(0,1)*41 .
 XIT       NAME                           .
           END                            .
  .
 BIN*      RES       1                    . CONTAINS THE BINARY RESULT OF SCAN IF
                                          . 'FIELD' CONTAINS A NUMERIC VALUE
 BYTE      +         1,0                  . CURRENT BYTE POSITION OF SCAN
 CARD*     RES       15                   . CARD INPUT AREA
 DATE*     +         '      '             . YYMMDD
           +         '      '             . DD MMM YY
           +         '      '             .
 FIELD*    RES       14                   . CONTAINS FIELDATA RESULT OF SCAN
 HOLD      RES       2                    . TEMPORARY STORAGE FOR 'MOVBIN'
 INFO*     RES       1                    . SCAN INFORMATION FLAGS:
                                          . T1 = LENGTH OF THE FD VALUE
                                          . S3 = TYPE (1-4)
                                          .       1 = ALPHA-NUMERIC
                                          .       2 = NUMERIC
                                          .       3 = LITERAL
                                          .       4 = BLANK (NULL) FIELD
                                          . S4 = EOC (END OF CARD)
                                          .       0 = NOT END OF CARD
                                          .       1 = END OF CARD
                                          . S5 = PNUM (SUBSCRIPT OF PUNCT IN PLIST)
                                          . S6 = PUNCT (PUNCTUATION MARK)
                                          .
 MON       +         'JAN'                . MONTH TABLE FOR 'DATETIME' SUBROUTINE
           +         'FEB'                .
           +         'MAR'                .
           +         'APR'                .
           +         'MAY'                .
           +         'JUN'                .
           +         'JUL'                .
           +         'AUG'                .
           +         'SEP'                .
           +         'OCT'                .
           +         'NOV'                .
           +         'DEC'                .
 NORMP     ARG       9,0,PLIST            . NORMAL PUNCTUATION LIST CONTROL
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...
NORMS    ARG      0,0,CARD      . NORMAL STARTING POSITION FOR SCAN
NPCT     +        9             . CURRENT PUNCTUATION LIST LENGTH
NUMTST   +        057           . LOW END OF NUMERIC TEST
         +        071           . HIGH END OF NUMERIC TEST
PCTL     +        1,PLIST       . CURRENT PUNCTUATION LIST CONTROL
PLIST    +        ','           . NORMAL PUNCTUATION LIST
         +        '='           .
         +        '/'           .
         +        '('           .
         +        ')'           .
         +        '.'           .
         +        '-'           .
         +        '+'           .
         +        ','           .
         +        072           . QUOTE MARK FOR LITERAL SCAN
QUOTE    RES      1             . +1 OR -1
SIGN     ARG      0,0,CARD      . STARTING POSITION FOR SCANNING
STARTC   RES      123           . REGISTER SAVE AREA
SV       RES      1             . T1 = NUMERIC DIGIT TEST AND COUNT
TEST                            . T2 = PLUS SIGN TEST
                                . T3 = MINUS SIGN TEST
                                .
         RES      1             . HH:MM:SS
TIME.    +        .     .       .
         +        .     .       . CURRENT WORD POSITION OF SCAN
WORD     +        1,0
  .
  .
$(1)     ERU$     .
/.       BINFD SUBROUTINE... THIS ROUTINE CONVERTS A SIGNED BINARY VALUE IN
  .                          MEMORY TO A FIELDATA (FD) CHARACTER STRING.
  .                          THE RESULT MAY BE ZERO SUPPRESSED WITH THE SIGN
  .                          FLOATED TO THE FIRST POSITION PRECEDING THE MOST
  .                          SIGNIFICANT DIGIT IF ENTRY IS MADE VIA 'ZERO'.
  .                          OTHERWISE, THE FDVAL HAS LEADING ZEROS WITH THE
  .                          SIGN (SPACE OR '-') IN THE FIRST POSITION OF THE
  .                          FIELD.
  .
  .                          ARG FORM  12,6,18
  .      INITIALIZATION...
  .                          LMJ  X11,BINFD   OR   LMJ  X11,ZERO
  .      CALL...             ARG  SC,0,FDVAL
  .                          ARG  NC,X,BINVAL
  .
  .                          SC         STARTING CHARACTER WITHIN FDVAL
  .      RESULT...           FDVAL      RESULTING FIELDATA VALUE STARTING LOC
  .                          NC         LENGTH OF THE FIELD (FDVAL)
  .                          X          X-REGISTER IF BINVAL IS INDEXED
  .                          BINVAL     LOCATION OF THE BINARY VALUE
  .
  .
  .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
BINFD*    SAVER,1    R3                 . SAVE R REGISTER
          LR         R3,(+' ')          . R3 = SPACE
          SR         R3,TEST            . TEST = FLAG FOR NO ZERO SUPPRESSION
          LR         R3,(+'0')          . R3 = FIELDATA ZERO
          SR         R3,TEST+1          . TEST+1 CONTAINS POSITIVE SIGN
          J          SVA                . SKIP ALTERNATE ENTRY POINT
ZERO*     SAVER,1    R3                 . SAVE R REGISTER
          SZ         TEST               . TEST = ZERO SUPPRESSION FLAG
          LR         R3,(+' ')          . R3 = SPACE
          SR         R3,TEST+1          . TEST+1 = POSITIVE SIGN
SVA       SAVEA,1    A2,A3,A4,A5,A6     . SAVE A REGISTERS
          SAVEX,1    X6,X7,X8           . SAVE X REGISTERS
          LR,T1      R3,1,X11           . R3 = NC (FIELD LENGTH)
          LA,T1      A4,0,X11           . A4 = SC
          AA         A4,R3              . A4 = SC + NC
          ANA,U      A4,2               . A4 = A4 - 2
          DSL        A4,36              . SHIFT FOR DIVIDE
          DI,U       A4,6               . A4 = WORD, A5 = BYTE
          AA,H2      A4,0,X11           . A4= FDVAL + WORD
          LX         X8,A4              . MOVE A4 TO X8 FOR 'PUTCR'
          LXI        X8,(-1)            . INCREMENT = -1 FOR REVERSE STORE
          LA,U       A3,5               . A3 = 5
          ANA        A3,A5              . A3 = 5 - BYTE
          LXI,U      A3,1               . INCREMENT = 1
          LA         A4,*1,X11          . A4 = BINARY VALUE
          LA         A5,TEST+1          . A5 = POSITIVE SIGN (' ' OR '0')
          TG,U       A4,0               . IS A4 NEGATIVE
          J          BNFD1              . NO, GO CONVERT THE BINARY VALUE
          LNA        A4,A4              . COMPLEMENT THE NUMBER IN A4
          LA         A5,(+'-')          . A5 = MINUS SIGN
BNFD1     SA         A5,SIGN            . SIGN = SPACE OR '-'
          J          ZRO1-2             . GO TO LOOP CONTROL
          DSL        A4,36              . SHIFT FOR DIVIDE
          DI,U       A4,10              . A4 = RESULT, A5 = REMAINDER
          OR,U       A5,060             . MASK ON THE ZONE BITS
          EX         PUTCR,*A3          . STORE THIS CHARACTER
          JZ         A4,ZRO1            . IF A4 = 0, THERE ARE NO MORE DIGITS
          JGD        R3,$-5             . LOOP BACK FOR ALL NC CHARACTERS
          J          ZRO3               . TRUNCATE HIGH IF NC CHARACTERS FILLED
ZRO1      TZ         TEST               . IS TEST = 0 (ZERO SUPPRESSION)
          J          ZRO2               . NO, GO FILL WITH LEADING ZEROS
          TNZ        R3                 . IS THERE ROOM FOR THE SIGN
          J          ZRO3               . NO, EXIT
          JGD        R3,$+1             . REDUCE COUNT BY 1
          LA         A6,SIGN            . A6 = SIGN (' ' OR '-')
          EX         PUTCR,*A3          . STORE IT IN FRONT OF THE FIRST DIGIT
          LA         A6,(+' ')          . REPLACE SIGN WITH A SPACE
          JGD        R3,$-2             . LOOP BACK TO STORE LEADING SPACES
          J          ZRO3               . EXIT AFTER FILLING NC CHARACTERS
ZRO2      LA         A6,(+'0')          . A6 = '0' FOR ZERO FILL
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
                TNZ        R3                  . IS THERE ROOM FOR LEADING ZEROS
                J          ZR03                . NO, EXIT
                JGD        R3,$+2              . REDUCE COUNT BY ONE AND LOOP
                EX         PUTCR,*A3           . STORE A FIELDATA ZERO
                JGD        R3,$-1              . LOOP FOR REMAINDER OF FIELD
                LA         A6,SIGN             . A6 = SIGN ('0' OR '-')
                TZ         R3                  . IS THERE ROOM FOR THE SIGN
                EX         PUTCR,*A3           . YES, STORE SIGN IN FIRST POSITION
        ZR03    RESTA,1    A2,A3,A4,A5,A6      . RESTORE A REGISTERS
                RESTX,1    X6,X7,X8            . RESTORE X REGISTERS
                RESTR,1    R3                  . RESTORE R REGISTER
                J          2,X11               . EXIT TO MAIN PROGRAM
        /.      BINOCT SUBROUTINE...THIS ROUTINE CONVERTS THE CONTENTS OF ONE WORD OF
        .                           STORAGE TO ITS 1-12 DIGIT OCTAL EQUIVALENT IN FD
        .                           FOR PRINTING.
        .
        .       INITIALIZATION...   ARG  FORM   12,6,18
        .
        .       CALL...             LMJ  X11,BINOCT
        .                           ARG  SC,O,FDVAL
        .                           ARG  NC,X,BINVAL
        .
        .       RESULTS...          SC         STARTING CHARACTER IN FDVAL ARRAY
        .                           FDVAL      1ST WORD OF RECEIVING ARRAY
        .                           NC         NUMBER OF CHARACTERS IN RECEIVING AREA
        .                           X          X-REGISTER IF BINVAL IS INDEXED
        .                           BINVAL     BINARY VALUE TO BE CONVERTED
        .
        .
        BINOCT* SAVEA      A3,A4,A5,A6         . SAVE A REGISTERS
                SAVEX      X6,X8               . SAVE X REGISTERS
                SAVER      R3                  . SAVE R REGISTER
                LA,T1      A4,0,X11            . A4 = SC
                LR,T1      R3,1,X11            . R3 = NC (FIELD LENGTH)
                AA         A4,R3               . A4 = SC + NC
                ANA,U      A4,2                . A4  A4 - 2
                DSL        A4,36               . SHIFT FOR DIVIDE
                DI,U       A4,6                . A4 = RELATIVE WORD, A5 = BYTE (0-5)
                AA,H2      A4,0,X11            . A4 = FDVAL + WORD
                LX         X8,A4               . MOVE A4 TO X8
                LXI        X8,(-1)             . SET INCREMENT TO -1
                LA,U       A3,5                . A3 = 5
                ANA        A3,A5               . A3 = 5 - BYTE
                LXI,U      A3,1                . INCREMENT = 1
                LA         A4,*1,X11           . A4 = BINVAL,X  (BINARY VALUE)
                J          $+6                 . GO TO LOOP CONTROL
                DSL        A4,3                . SHIFT 3 BITS TO A5
                SSL        A5,3                . PUT 3 LEADING ZERO BITS ON A5
                LSSC       A5,6                . ROTATE TO LOW END
                OR,U       A5,060              . MASK ON ZONE BITS (BIN TO FD)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
                 EX       PUTCR,*A3          . STORE RIGHT TO LEFT FROM A6
                 JGD      R3,$-5             . LOOP BACK FOR ALL 12 DIGITS
                 RESTA    A3,A4,A5,A6        . RESTORE A REGISTERS
                 RESTX    X6,X8              . RESTORE X REGISTERS
                 RESTR    R3                 . RESTORE R REGISTER
                 J        2,X11              . RETURN
   /.            DATETIME SUBROUTINE..THIS ROUTINE PICKS UP THE CURRENT DATE AND TIME
    .                                 OF DAY FROM EXEC 8 AND CONVERTS THEM TO A FORM
    .                                 SUITABLE FOR PRINTING.
    .
    .            INITIALIZATION...    NONE
    .
    .            CALL...              LMJ  X11,DATETIME
    .
    .            RESULTS...           DATE      YYMMDD
    .                                 DATE+1    DD MMM YY
    .                                 TIME      HH:MM:SS
    .
    .
   DATETIME* SAVEA    A0,A1              . SAVE A REGISTERS
             ER       DATE$              . A0 = MMDDYY
             SA,S3    A0,DATE+2          . 2ND DIGIT OF YEAR
             SSC      A0,6               . A0 = YMMDDY
             SA,S2    A0,DATE+2          . 1ST DIGIT OF YEAR
             SSC      A0,6               . MOVE YY TO FRONT
             SA       A0,DATE            . DATE = YYMMDD
             SA,T1    A0,DATE+1          . STORE DD PART OF 'DD MMM YY'
             LA       A0,('  :  :')      . A0 = MASKWORD FOR TIME, A1 = HHMMSS
             LDSC     A0,12              . A0 = :  :HH    A1 = MMSS
             LSSC     A0,6               . A0 =  :HH:     A1 = MMSS
             LDSC     A0,12              . A0 = :HH:MM    A1 = SS
             LSSC     A0,6               . A0 = HH:MM:    A1 = SS
             DS       A0,TIME            . STORE IT AWAY
             ER       TDATE$             . A0 = MDYTIM (BINARY)
             SSL      A0,30              . A0 = MM (INDEX FOR MONTH TABLE)
             LA       A1,MON-1,A0        . A1 = MMM (FIELDATA)
             SA,H2    A1,DATE+1          . STORE MMM PART OF 'DD MMM YY'
             RESTA    A0,A1              . RESTORE A REGISTERS
             J        0,X11              .RETURN
   /.        FDBIN SUBROUTINE... THIS ROUTINE CONVERTS A FIELDATA (FD) CHARACTER
    .                            STRING TO A SIGNED BINARY VALUE.  IF THE INPUT
    .                            VALUE HAS BEEN PRE-EDITED OR IS KNOWN TO BE A
    .                            VALID NUMERIC QUANTITY, THE EDIT FLAG (E) MUST
    .                            BE SET TO ZERO.
    .
    .        INITIALIZATION...   ARG FORM  12,6,18
    .
    .        CALL...             LMJ  X11,FDBIN
    .                            ARG  SC,E,FDVAL
    .                            ARG  NC,X,BINVAL
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
                                  +     ERRXIT        NOT PRESENT IF E=0
.
.                                 SC        STARTING CHARACTER WITHIN FDVAL
.          RESULT...              E         O=NOEDIT, 1=EDIT(ERRXIT REQUIRED)
.                                 FDVAL     FIELDATA VALUE STARTING ADDRESS
.                                 NC        LENTH OF FDVAL
.                                 X         X-REGISTER IF BINVAL IS INDEXED
.                                 BINVAL    RESULT OF CONVERSION
.                                 ERRXIT    IF E=1 THIS IS THE ERROR EXIT.
.                                           OTHERWISE, IT IS NOT CODED.  IF THE
.                                           ERROR EXIT IS TAKEN, BINVAL CONTAINS
.                                           ONE OF THE FOLLOWING VALUES.
.
.                                           0 = FDVAL WAS ALL SPACES OR SIGN ONLY
.                                           1 = NON-NUMERIC CHARACTER FOUND
.                                           2 = SIGN DOES NOT PRECEDE VALUE
.                                           3 = NUMBER IS TOO LARGE FOR 1 WORD
.
.
.
FDBIN*     SAVEA,2     A0,A1,A2,A4,A5,A13,A14 . SAVE A REGISTERS
           SAVEX,2     X6,X7                 . SAVE X REGISTERS
           SAVER,2     R3                    . SAVE R REGISTER
           SZ          TEST                  . INITIALIZE TEST WORD
           LR,T1       R3,1,X11              . R3 = NC (LENGTH OF FIELD)
           LA,T1       A1,0,X11              . A1 = SC
           ANA,U       A1,1                  . A1 = SC - 1
           DSL         A1,36                 . A1 = 0, A2 = SC-1
           DI,U        A1,6                  . A1 = RELATIVE WORD, A2 = BYTE
           AA,H2       A1,0,X11              . A1 = ACTUAL WORD ADDRESS
           LX          X7,A1                 . X7 = WORD ADDRESS
           LXI,U       A2,1                  . A2(35-18) = 1
           LXI,U       X7,1                  . X7(35-18) = 1
           SZ          A1                    . A1 = 0    (RESULTING BINARY VALUE)
           LA,U        A0,1                  . A0 = 1    (CONSTANT SOURCE OF 1)
           SA          A0,SIGN               . SIGN = +1
           DL          A13,NUMTST            . SET A13 AND A14 TO NUMERIC TEST VALUES
           J           FDBN3                 . GO TO LOOP CONTROL
FDBN1      EX          GETC,*A2              . GET 1 CHARACTER FROM STRING
           TNE         A4,(+' ')             . IS IT A BLANK
           J           BLANK                 . YES, GO SEE IF IT IS LEADING OR TRAILING
           TNZ,S3      0,X11                 . ARE WE GOING TO EDIT THIS FIELD
           J           FDBN2                 . NO, ASSUME VALID DATA
           TNE         A4,(+'+')             . IS IT A PLUS SIGN
           J           PLUS                  . GO SEE IF IT IS VALID
           TNE         A4,(+'-')             . IS IT A MINUS SIGN
           J           MINUS                 . GO SEE IF IT IS VALID
           TW          A13,A4                . IS THIS A NUMERIC CHARACTER
           J           E1                    . NO, ERROR CODE 1
FDBN2      LA,T1       A5,TEST               . A5 = NUMERIC DIGIT COUNT
           AA          A5,A0                 . A5 = A5 + 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
         SA,T1      A5,TEST              . STORE NEW COUNT
         AND,U      A4,017               . STRIP OFF ZONE BITS
         MSI,U      A1,10                . MULTIPLY PREVIOUS VALUE BY 10
         AA         A1,A5                . ADD IN THIS DIGIT
FDBN3    JGD        R3,FDBN1             . LOOP BACK FOR ALL NC DIGITS
         MSI        A1,SIGN              . SIGN THE BINARY VALUE
         TNZ,S3     0,X11                . HAVE WE BEEN EDITING
         J          FDBN4+1              . NO, SKIP NEXT TEST
         TNZ,T1     TEST                 . DID WE FIND AT LEAST 1 DIGIT
         J          FDBN4                . NO, TAKE ERROR EXIT WITH A1 UNCHANGED
         LA,U       A0,10                . A0 = 10
         TG,T1      A0,TEST              . IS TEST .GT. 10
         J          FDBN4+1              . NO, GO STORE VALUE
         LA,U       A1,3                 . A1 = ERROR CODE 3
FDBN4    LA,U       A2,99                . A2 = ERROR CONDITION SWITCH
         RESTX,2    X6,X7                . RESTORE X REGISTERS
         RESTR,2    R3                   . RESTORE R REGISTER
         SA         A1,*1,X11            . STORE THE BINARY RESULT (OR ERROR)
         TZ,S3      0,X11                . HAVE WE BEEN EDITING
         J          EXIT1                . YES, GO SEE IF WE FOUND ANY ERRORS
         RESTA,2    A0,A1,A2,A4,A5,A13,A14 . RESTORE A REGISTERS
         J          2,X11                . E=0, NORMAL EXIT
EXIT1    TNE,U      A2,99                . DID WE FIND AN ERROR
         J          EXIT2                . YES, GO TO NEXT EXIT ROUTINE
         RESTA,2    A0,A1,A2,A4,A5,A13,A14 . RESTORE A REGISTERS
         J          3,X11                . E=1, NORMAL EXIT
EXIT2    RESTA,2    A0,A1,A2,A4,A5,A13,A14 . RESTORE A REGISTERS
         J          *2,X11               . E=1, ERROR EXIT
.
BLANK    TNZ,T1     TEST                 . SEE IF AT LEAST 1 DIGIT PICKED UP
         J          FDBN3                . BYPASS LEADING BLANKS
         J          FDBN3+1              . TERMINATE ON A TRAILING BLANK
PLUS     TZ,T2      TEST                 . HAVE WE BEEN HERE BEFORE
         J          E2                   . YES, ERROR CODE 2
         TZ,T1      TEST                 . HAVE WE PICKED UP AT LEAST 1 DIGIT
         J          E2                   . YES, ERROR CODE 2
         SA,T2      A0,TEST              . SET SWITCH FOR NEXT ENTRY
         J          FDBN3                . GO TO LOOP CONTROL
MINUS    TZ,T3      TEST                 . HAVE WE BEEN HERE BEFORE
         J          E2                   . YES, ERROR CODE = 2
         TZ,T1      TEST                 . HAVE WE PICKED UP AT LEAST 1 DIGIT
         J          E2                   . YES, ERROR CODE = 2
         SA,T3      A0,TEST              . SET SWITCH FOR NEXT ENTRY
         SNA        A0,SIGN              . SIGN = -1
         J          FDBN3                . GO TO LOOP CONTROL
.
E1       LA,U       A1,1                 . A1 = ERROR CODE 1
         J          FDBN4                . GO SET ERROR FLAG
E2       LA,U       A1,2                 . A1 = ERROR CODE 2
         J          FDBN4                . GO SET ERROR FLAG
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
 /.        MOVBIN SUBROUTINE...THIS ROUTINE CONVERTS A SIGNED BINARY VALUE TO
 .                             FIELDATA (FD) AND STORES ALL SIGNIFICANT DIGITS
 .                             (PLUS THE MINUS SIGN FOR NEGATIVE NUMBERS) LEFT
 .                             TO RIGHT BEGINNING WITH SC OF FDVAL.  THE LENGTH
 .                             OF THE CHARACTER STRING IS RETURNED IN NC.
 .
 .         INITIALIZATION...   ARG   FORM  12,6,18
 .
 .         CALL...             LMJ   X11,MOVBIN
 .                             ARG   SC,0,FDVAL
 .                             ARG   NC,X,BINVAL
 .
 .         RESULTS..           SC          STARTING POSITION OF FDVAL
 .                             FDVAL       1ST WORD OF RECEIVING AREA
 .                             NC          LENGTH OF CHARACTER STRING FROM MOVBIN
 .                             X           X-REGISTER IF BINVAL IS INDEXED
 .                             BINVAL      BINARY VALUE
 .
 .
 MOVBIN*   SAVEA      A2,A3,A4,A15         . SAVE A REGISTERS
           SAVEX      X6,X7,X8,X11         . SAVE X REGISTERS
           SAVER      R3                   . SAVE R REGISTER
           LA,T1      A15,0,X11            . A15 = SC
           ANA,U      A15,1                . A15 = SC - 1
           DSL        A15,36               . SHIFT FOR DIVIDE
           DI,U       A15,6                . A15 = WORD, A16 = BYTE (0-5)
           AA,H2      A15,0,X11            . A15 = FDVAL + WORD
           AA         A15,(1,0)            . A15(35-18) = 1
           LA         A2,*1,X11            . A2 = BINARY VALUE TO BE CONVERTED
           SA         A2,SIGN              . STORE IT AWAY FOR 'ZERO'
           LMJ        X11,ZERO             . BINFD WITH ZERO SUPPRESSION AND SIGN
           ARG        1,0,HOLD             . RECEIVING AREA (12 CHARACTERS)
           ARG        12,0,SIGN            . LENGTH OF TEST AND LOCATION OF BINVAL
           LX         X7,(1,HOLD)          . X7 = LOCATION OF FD RESULT
           LA         A2,(1,0)             . A2 = 1ST BYTE WITHIN 'TEST'
           LX         X8,A15               . X8 = REQUESTED FD AREA (FDVAL)
           LA         A3,034               . A3 = BYTE WITHIN FDVAL
           LXI,U      A3,1                 . A3 INCREMENT = 1
           LR,U       R3,11                . R3 = 11 TO LOOP 12 TIMES
           SZ         A15                  . A15 = 0 (TO COMPUTE NC)
           EX         GETC,*A2             . GET A CHARACTER
           TNE        A4,(+' ')            . IS IT A BLANK
           J          $+3                  . YES, IGNORE IT
           EX         PUTC,*A3             . MOVE CHARACTER TO FDVAL
           AA,U       A15,1                . COUNT THE CHARACTERS MOVED
           JGD        R3,$-5               . LOOP FOR ALL 12 CHARACTERS
           RESTX      X6,X7,X8,X11         . RESTORE X REGISTERS
           SA,T1      A15,1,X11            . STORE THE LENGTH OF THE FIELD
           RESTA      A2,A3,A4,A15         . RESTORE A REGISTERS
           RESTR      R3                   . RESTORE R REGISTER
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
          J          2,X11              . RETURN TO CALLING PROGRAM
/.        MOVFD SUBROUTINE... THIS ROUTINE MOVES A FIELDATA (FD) CHARACTER
.                             STRING LEFT TO RIGHT FROM ONE AREA TO ANOTHER.
.
.         INITIALIZATION...   ARG  FORM  12,6,18
.
.         CALL...             LMJ  X11,MOVFD
.                             ARG  NC,BYTE,WORD       'FROM'
.                             ARG  NB,BYTE,WORD       'TO'
.
.         RESULT...           NC        LENGTH OF THE CHARACTER STRING
.                             BYTE      1ST BYTE WITHIN THE WORD (0-5)
.                             WORD      1ST WORD OF THE FIELD
.                             NB        NUMBER OF BLANKS (SPACES) TO MOVE
.                                       FOLLOWING THE CHARACTER STRING MOVE
.
MOVFD.    SZ         TEST               . TEST = 0 FOR MOVFD
          J          $+2                . JUMP TO SAVEA PROC CALL
MOVFD1.   SX         X11,TEST           . TEST = NONZERO FOR MOVFD1
          SAVEA      A2,A4,A3           . SAVE A REGISTERS
          SAVEX      X6,X7,X8           . SAVE X REGISTERS
          SAVER      R3                 . SAVE R REGISTER
          LR,T1      R3,0,X11           . R3 = LENGTH OF CHARACTER STRING
          LA,S3      A2,0,X11           . A2 = BYTE OF 'FROM' FIELD
          LXI,U      A2,1               . A2 INCREMENT = 1
          LX,H2      X7,0,X11           . X7 = WORD ADDRESS OF 'FROM'
          LXI,U      X7,1               . X7 INCREMENT = 1
          LA,S3      A3,1,X11           . A3 = BYTE OF 'TO' FIELD
          LXI,U      A3,1               . A3 INCREMENT = 1
          LX,H2      X8,1,X11           . X8 = WORD ADDRESS OF 'TO'
          LXI,U      X8,1               . X8 INCREMENT = 1
          J          $+3                . GO TO LOOP CONTROL
          EX         GETC,*A2           . GET A CHARACTER
          EX         PUTC,*A3           . STORE IT
          JGD        R3,$-2             . LOOP FOR ALL NC CHARACTERS
          LR,T1      R3,1,X11           . R3 = NB (NUMBER OF BLANKS TO BE MOVED)
          LA,U       A4,+' '            . A4 = SPACE
          J          $+2                . GO TO LOOP CONTROL
          EX         PUTC,*A3           . STORE A BLANK
          JGD        R3,$-1             . LOOP FOR ALL NB CHARACTERS
          TZ         TEST               . DID WE ENTER THROUGH MOVFD
          J          MOVFD2             . NO, TAKE ALTERNATE EXIT
          RESTA      A2,A4,A3           . RESTORE A REGISTERS
          RESTX      X6,X7,X8           . RESTORE X REGISTERS
          J          MOVFD3             . MOVE TO EXIT ROUTINE
MOVFD2    RESTA      A2,A4              . RESTORE A REGISTERS (EXCEPT A3)
          RESTX      X6,X7              . RESTORE X REGISTERS (EXCEPT X8)
MOVFD3    RESTR      R3                 . RESTORE R REGISTER
          J          2,X11              . RETURN TO CALLING PROGRAM
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
/.        OCTBIN SUBROUTINE...THIS ROUTINE CONVERTS A FIELDATA (FD) CHARACTER
 .                            STRING TO A BINARY VALUE.  EACH DIGIT WILL BE
 .                            EDITED FOR VALIDITY (0-7).  ANY OUT OF RANGE
 .                            CHARACTER WILL TERMINATE THE SCAN AND THE BINARY
 .                            VALUE COLLECTED TO THAT POINT WILL BE SAVED.
 .
 .        INITIALIZATION...   ARG FORM  12,6,18
 .
 .        CALL...             LMJ  X11,OCTBIN
 .                            ARG  SC,0,FDVAL
 .                            ARG  NC,0,BINVAL
 .
 .        RESULTS...          SC        STARTING CHARACTER WITHIN FDVAL
 .                            FDVAL     FIELDATA VALUE STARTING ADDRESS
 .                            NC        LENGTH OF FDVAL
 .                            BINVAL    RESULT OF CONVERSION
 .
 .
 .
   OCTBIN.   SAVEA    A2,A3,A4,A13,A14   . SAVE A REGISTERS
             SAVEX    X6,X7              . SAVE X REGISTERS
             SAVER    R3                 . SAVE R REGISTER
             LA,T1    A3,0,X11           . A3 = SC
             LR,T1    R3,1,X11           . R3 = NC
             ANA,U    A3,1               . A3 = A3 - 1
             DSL      A3,36              . SHIFT FOR DIVIDE
             DI,U     A3,6               . A3 = RELATIVE WORD, A4 = BYTE
             AA,H2    A3,0,X11           . A3 = ACTUAL WORD ADDRESS
             LA       A2,A4              .
             LXI,U    A2,1               . A2 = (1,WORD)
             LX       X7,A3              .
             LXI,U    X7,1               . X7 = (1,BYTE)
             LA,U     A13,057            .
             LA,U     A14,067            . A13-A14 = RANGE CHECK
             SZ       A3                 . INITIALIZE BINARY VALUE
             J        OB2                . GO TO LOOP CONTROL
   OB1       EX       GETC,*A2           . GET 1 CHARACTER FROM STRING
             TW       A13,A4             . IS IT WITHIN RANGE (0-7)?
             J        OB2+1              . NO, EXIT
             SSC      A4,3               . YES, MOVE LOW 3 BITS TO 35-33
             LDSL     A3,3               . SHIFT 3 BITS ONTO END OF PREVIOUS VALUE
   OB2       JGD      R3,OB1             . LOOP FOR ALL NC CHARACTERS
             RESTX    X6,X7              . RESTORE X REGISTERS
             RESTR    R3                 . RESTORE R REGISTERS
             SA       A3,*1,X11          . STORE BINARY VALUE
             RESTA    A2,A3,A4,A13,A14   . RESTORE A REGISTERS
             J        2,X11              . RETURN
/.          ISCAN SUBROUTINE... THIS ROUTINE INITIALIZES THE POINTERS FOR 'SCAN'
 .                              AND MUST BE CALLED BEFORE ANY CALL TO 'SCAN'.  IT
 .                              WILL PERFORM THE FIRST CALL TO 'SCAN' FOR THE
 .                              CURRENT SCAN AREA BEFORE RETURNING.
 .
```

516

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

```
LINE #   TEXT...

   .
   .          INITIALIZATION...   NORMALLY ONE CALL TO 'SETSCN' WILL BE MADE PRIOR
   .                              TO PERFORMING 'ISCAN'.  IF IT IS NOT, THE NORMAL
   .                              SCANNING PARAMETERS WILL BE USED.
   .
   .          CALL...             LMJ  X11,ISCAN
   .                              +    CLENGTH
   .
   .          RESULTS...          CLENGTH   LENGTH OF THE CURRENT SCAN AREA IN WORDS
   .
   .                              SINCE 'ISCAN' ACTUALLY PERFORMS THE FIRST SCAN,
   .                              THE RESULTS WILL BE THE SAME AS A CALL TO 'SCAN'.
   .
 ISCAN*    SAVEA     A3,A15              . SAVE A REGISTERS
           LA        A15,STARTC          . A15 = STARTING POSITION
           SA,H2     A15,WORD            . SET THE WORD ADDRESS
           SSL       A15,24              . MOVE THE BYTE OVER TO THE RIGHT
           SA,H2     A15,BYTE            . SET THE BYTE NUMBER WITHIN WORD
           LA,H2     A3,*0,X11           . A3 = LENGTH OF THE CURRENT SCAN AREA
           AA,H2     A3,WORD             . COMPUTE LAST WORD OF AREA + 1
           LA,U      A15,077             . A15 = SCAN TERMINATING CHARACTER
           SA,S1     A15,0,A3            . STORE THE TERMINATOR CHARACTER
           RESTA     A3,A15              . RESTORE A REGISTERS
           AX,U      X11,1               . BUMP X11 BY 1 SO 'SCAN' RETURNS TO ONE
                                         . WORD PAST THE CLENGTH PARAMETER
           J         SCAN                . GO PERFORM THE FIRST SCAN OF THIS AREA
 /.        SCAN SUBROUTINE...  THIS ROUTINE PERFORMS A FREE FORMAT SCAN OF THE
   .                           CURRENT SCAN AREA BEGINNING WITH THE POSITION OF
   .                           THE LAST CALL TO 'SCAN' OR 'ISCAN'.
   .       INITIALIZATION...   THE SCAN POINTERS MUST BE INITIALIZED BY A CALL
   .                           TO 'ISCAN', THEN, ALL SUBSEQUENT CALLS TO 'SCAN'
   .                           WILL LEAVE THE POINTERS SET FOR THE NEXT ENTRY.
   .       CALL...             LMJ  X11,SCAN
   .
   .       RESULTS...          FIELD     FIELDATA VALUE LEFT JUSTIFIED WITH
   .                                     TRAILING BLANKS
   .                           BIN       BINARY VALUE IF NTYP = 3
   .                           INFO      SCAN INFORMATION
   .
   .                                     T1  LENGTH OF FD VALUE IN FIELD
   .                                     S3  TYPE OF FIELD
   .                                     S4  END OF CARD FLAG
   .                                     S5  PNUM (SUBSCRIPT IN PLIST)
   .                                     S6  PUNCT (PUNCTUATION MARK)
   .
   .                           SEE D-BANK ENTRIES FOR A MORE COMPLETE PICTURE
   .                           OF THE CONTENTS OF 'INFO'.
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

SUBRTN DICK ROGERS' UTILITY SUBROUTINE PACKAGE

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...

         .
         SCAN*    SAVEA     A2,A3,A4,A13,A14,A15,A12 . SAVE A REGISTERS
                  SAVEX     X6,X7,X8,X9         . SAVE X REGISTERS
                  SAVER     R1                  . SAVE R REGISTER
                  SZ        BIN                 . SET BINARY VALUE TO ZERO
                  SZ        INFO                . ZERO OUT THE INFORMATION AREA
                  LR,U      R1,14               . R1 = LENGTH OF 'FIELD'
                  LX        X7,(0,('      '))   . SENDING CONTROL WORD
                  LX        X8,(1,FIELD)        . RECEIVING CONTROL WORD
                  BT        X8,0,*X7            . CLEAR 'FIELD' TO SPACES
                  DL        A13,NUMTST          . SET A13 AND A14 TO NUMERIC TEST VALUES
                  LA,U      A12,2               . A12 = 2 (NUMERIC FLAG)
                  LX        X7,WORD             . X7 = WORD ADDRESS OF THIS SCAN
                  LA        A2,BYTE             . A2 = BYTE WITHIN WORD
                  LX        X8,(1,FIELD)        . X8 = 1ST WORD OF RECEIVING AREA
                  LA        A3,(1,0)            . A3 = BYTE WITHIN 'FIELD'
                  SZ        A15                 . A15 = CHARACTER COUNT
                  EX        GETC,*A2            . GET A CHARACTER
                  TNE       A4,(+' ')           . IS IT A SPACE
                  J         $-2                 . YES, BYPASS LEADING SPACES
                  TNE       A4,QUOTE            . IS THIS A QUOTE
                  J         LITERL              . YES, GO TO THE LITERAL SCAN ROUTINE
         SC1      TNE,U     A4,077              . IS THIS THE TERMINATOR CHARACTER
                  J         ENDC                . YES, END OF CARD
                  LR        R1,NPCT             . R1 = NUMBER OF PUNCTUATION MARKS
                  LX        X9,PCTL             . X9 = 'PLIST' CONTROL
                  SE        A4,0,*X9            . IS THIS A PUNCTUATION MARK
                  J         $+2                 . NO, GO SEE IF IT IS NUMERIC
                  J         ENDF                . YES, GO TO END OF FIELD ROUTINE
                  TW        A13,A4              . IS IT A NUMERIC CHARACTER
                  LA,U      A12,1               . NO, RESET FLAG TO 'ALPHANUMERIC'
                  EX        PUTC,*A3            . STORE THE CHARACTER IN 'FIELD'
                  AA,U      A15,1               . COUNT IT
                  EX        GETC,*A2            . PICK UP THE NEXT CHARACTER
                  TE        A4,(+' ')           . IS IT A SPACE
                  J         SC1                 . NO, GO SEE WHAT IT IS
         RUNOUT   SX        X7,WORD             . SAVE THE CURRENT SCAN POSITION
                  SA,H2     A2,BYTE             . SAVE MORE SCAN INFO
                  EX        GETC,*A2            . PICK UP THE NEXT CHARACTER
                  TNE       A4,(+' ')           . IS THIS A SPACE
                  J         $-2                 . YES, BYPASS TRAILING SPACES
                  TNE,U     A4,077              . IS THIS THE TERMINATOR
                  J         ENDFC               . YES, END OF FIELD AND END OF CARD
                  TNE       A4,QUOTE            . IS THIS A QUOTE MARK
                  J         ENDFB               . END OF FIELD WITH BLANK AS PUNCTUATION
                  LR        R1,NPCT             . R1 = LENGTH OF 'PLIST'
                  LX        X9,PCTL             . X9 = PUNCTUATION LIST CONTROL
                  SE        A4,0,*X9            . IS THIS A PUNCTUATION MARK
                  J         ENDFB               . NO, END OF FIELD WITH BLANK PUNCTUATION
         ENDF     SX        X7,WORD             . YES, SAVE NEW SCAN POINTERS
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE # TEXT...

```
                SA,H2     A2,BYTE          . SAVE THE BYTE NUMBER, ALSO
                SA,S6     A4,INFO          . SAVE 'PUNCT'
                LA        A4,NPCT          . A4 = LENGTH OF 'PLIST'
                ANA       A4,R1            . SUBTRACT SEARCH INDEX
                SA,S5     A4,INFO          . SAVE 'PNUM'
                J         ALLEND           . GO TO NORMAL EXIT
       ENDFC    LA        A4,(010005)      . A4 = 'EOC', 'PNUM' AND 'PUNCT'
                SA,H2     A4,INFO          . STORE 'EM AWAY
                J         ALLEND           . GO TO NORMAL EXIT
       LITERL   LA,U      A12,3            . A12 = LITERAL FLAG
                EX        GETC,*A2         . GET THE NEXT CHARACTER
                TNE       A4,QUOTE         . IS IT A QUOTE MARK
                J         CKEND            . YES, COULD BE THE END OF THE LITERAL
                TNE,U     A4,077           . IS IT THE TERMINATOR
                J         ENDC             . YES, GO TO END OF CARD ROUTINE
       SC2      EX        PUTC,*A3         . STORE IT
                AA,U      A15,1            . COUNT IT
                J         LITERL+1         . LOOP BACK FOR MORE
       CKEND    EX        GETC,*A2         . LOOK AT NEXT CHARACTER
                TNE       A4,QUOTE         . IS IT ALSO A QUOTE
                J         SC2              . YES, DOUBLE QUOTE YIELDS ONE, STORE IT
                ANA,U     A2,1             . BACK UP ONE CHARACTER FOR 'RUNOUT'
                J         RUNOUT           . QUOTE/NON-QUOTE = END OF LITERAL
       ENDC     LA,U      A4,1             . A4 = EOC FLAG
                SA,S4     A4,INFO          . STORE IT IN 'INFO'
       ENDFB    LA,U      A4,+' '          . A4 = 'PNUM' AND 'PUNCT'
                SA,T3     A4,INFO          . STORE IN INFORMATION WORD
       ALLEND   SA,T1     A15,INFO         . STORE LENGTH OF FIELD
                SA,T1     A15,ARG2         . SAVE IT FOR 'FDBIN' ALSO
                TNZ       A15              . DID WE FIND AT LEAST ONE CHARACTER
                LA,U      A12,4            . NO, A12 = BLANK (NULL) FIELD FLAG
                SA,S3     A12,INFO         . SAVE THE 'TYPE' FLAG
                TE,U      A12,2            . IS THIS A NUMERIC FIELD
                J         EXITS            . NO, SKIP THE FIELDATA TO BINARY ROUTINE
                SX,H2     X11,EXITS-1      . SAVE X11
                LMJ       X11,FDBIN        . CONVERT 'FIELD' TO 'BIN'
                ARG       1,0,FIELD        . BEGIN WITH CHARACTER 1 OF 'FIELD'
       ARG2     ARG       0,0,BIN          . FIRST SUBFIELD FILLED IN WITH LENGTH
                LX,U      X11,0            . 'U' FILLED IN ABOVE
       EXITS    RESTA     A2,A3,A4,A13,A14,A15,A12 . RESTORE A REGISTERS
                RESTX     X6,X7,X8,X9      . RESTORE X REGISTERS
                RESTR     R1               . RESTORE R REGISTER
                J         0,X11            . RETURN TO CALLING PROGRAM
       /.       SETSCN SUBROUTINE...THIS ROUTINE SETS THE FREE FORMAT SCAN PARAMETERS
       .                            (STARTC, NPCT AND PCTL) FOR 'ISCAN' AND 'SCAN'.
       .
       .        INITIALIZATION...   ARG  FORM  12,6,18
       .
       .        CALL...             LMJ  X11,SETSCN
       .                            ARG  BYTE,0,WORD   OR    + 0
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...
         .                                  ARG  NPCT,0,PLIST  OR    + 0
         .
         .              RESULT...           BYTE      1ST BYTE WITHIN WORD (0-5)
         .                                  WORD      1ST WORD OF SCAN AREA
         .                                  NPCT      NUMBER OF PUNCTUATION MARKS
         .                                  PLIST     LOCATION OF PUNCTUATION MARK LIST
         .
         .                                  IF 1ST ARG = +0 SETSCN USES 0,0,CARD
         .                                  IF 2ND ARG = +0 SETSCN USES 9,0,PLIST
         .
         .
         SETSCN*   SAVEA      A15                   . SAVE A REGISTER
                   LA         A15,0,X11             . A15 = 1ST ARGUMENT
                   JN         A15,SETSCN1           . IGNORE 1ST ARGUMENT IF IT IS NEGATIVE
                   TNZ        A15                   . IS IT ZERO
                   LA         A15,NORMS             . YES, REPLACE WITH  0,0,CARD
                   SA         A15,STARTC            . STARTC = BYTE,U,WORD
         SETSCN1   LA         A15,1,X11             . A15 = 2ND ARGUMENT
                   JN         A15,SETSCN2           . IGNORE 2ND ARGUMENT IF IT IS NEGATIVE
                   TNZ        A15                   . IS IT ZERO
                   LA         A15,NORMP             . YES, REPLACE WITH 9,0,PLIST
                   SA,H2      A15,PCTL              . STORE PLIST ADDRESS
                   SSL        A15,24                . SHIFT NPCT TO THE RIGHT
                   SA         A15,NPCT              . STORE NPCT
         SETSCN2   RESTA      A15                   . RESTORE A REGISTER
                   J          2,X11                 . EXIT
         /.        GETC SUBROUTINE...  THIS ROUTINE PICKS UP ONE CHARACTER EACH TIME IT
         .                             IS CALLED.  IF THE NEXT CHARACTER IS (S1) OF THE
         .                             NEXT WORD, (X6) IS USED TO UPDATE THE REGISTERS.
         .
         .         INITIALIZATION...   LX   X7,(1,WORD)     1ST WORD OF AREA
         .                             LA   A2,(1,BYTE)     1ST BYTE WITHIN WORD (0-5)
         .
         .
         .         CALL...             EX   GETC,*A2
         .
         .         RESULT...           A4 = CHARACTER WITH LEADING ZEROS
         .                             X6 = ADDRESS OF EX + 1 ONLY IF WORD OVERFLOW
         .                                  OCCURED.  OTHERWISE, IT IS UNCHANGED.
         .
         .
         .
         GETC*     LA,S1      A4,0,X7               . BYTE 0 OF WORD
                   LA,S2      A4,0,X7               . BYTE 1 OF WORD
                   LA,S3      A4,0,X7               . BYTE 2 OF WORD
                   LA,S4      A4,0,X7               . BYTE 3 OF WORD
                   LA,S5      A4,0,X7               . BYTE 4 OF WORD
                   LA,S6      A4,0,X7               . BYTE 5 OF WORD
                   LMJ        X6,$+1                . BYTE 0 OF NEXT WORD
                   LXM,U      A2,1                  . A2 = BYTE 1 FOR NEXT CALL
                   LA,S1      A4,1,*X7              . COMPLETE THIS CALL (GET BYTE 0, NOW)
                   J          0,X6                  . RETURN TO ADDRESS OF EX + 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
/.          PUTC SUBROUTINE...  THIS ROUTINE STORES ONE CHARACTER EACH TIME IT IS
.                               CALLED.  IF THE NEXT STORAGE POSITION IS (S1) OF
.                               THE NEXT WORD, (X6) IS USED TO UPDATE THE
.                               REGISTERS.
.
.           INITIALIZATION...   LX    X8,(1,WORD)     1ST WORD OF AREA
.                               LA    A3,(1,BYTE)     1ST BYTE WITHIN WORD (0-5)
.
.           CALL...             EX    PUTC,*A3
.
.           RESULT...           A4 = CHARACTER WHICH WAS STORED
.                               X6 = ADDRESS OF EX + 1 ONLY IF WORD OVERFLOW
.                                    OCCURED.  OTHERWISE, IT IS UNCHANGED.
.
.
PUTC*       SA,S1     A4,0,X8           . BYTE 0 OF WORD
            SA,S2     A4,0,X8           . BYTE 1 OF WORD
            SA,S3     A4,0,X8           . BYTE 2 OF WORD
            SA,S4     A4,0,X8           . BYTE 3 OF WORD
            SA,S5     A4,0,X8           . BYTE 4 OF WORD
            SA,S6     A4,0,X8           . BYTE 5 OF WORD
            LMJ       X6,$+1            . BYTE 0 OF NEXT WORD
            LXM,U     A3,1              . A3 = BYTE 1 FOR NEXT CALL
            SA,S1     A4,1,*X8          . COMPLETE THIS CALL (STORE BYTE 0, NOW)
            J         0,X6              . RETURN TO LOCATION OF EX + 1
/.          PUTCR SUBROUTINE... THIS ROUTINE STORES ONE CHARACTER EACH TIME IT IS
.                               CALLED.  IF THE NEXT STORAGE POSITION (S6) OF THE
.                               NEXT WORD, (X6) IS USED TO UPDATE THE REGISTERS.
.                               NOTE THAT THE CHARACTERS ARE STORED RIGHT TO LEFT.
.
.           INITIALIZATION...   LX    X8,(-1,WORD)    LAST WORD OF AREA
.                               LA    A3,(1,5-BYTE)   LAST BYTE WITHIN WORD (0-5)
.
.           CALL...             EX    PUTCR,*A3
.
.           RESULT...           A6 = CHARACTER WHICH WAS STORED
.                               X6 = ADDRESS OF EX + 1 ONLY IF WORD OVERFLOW
.                                    OCCURED.  OTHERWISE, IT IS UNCHANGED.
.
.
PUTCR*      SA,S6     A6,0,X8           . BYTE 5 OF WORD
            SA,S5     A6,0,X8           . BYTE 4 OF WORD
            SA,S4     A6,0,X8           . BYTE 3 OF WORD
            SA,S3     A6,0,X8           . BYTE 2 OF WORD
            SA,S2     A6,0,X8           . BYTE 1 OF WORD
            SA,S1     A6,0,X8           . BYTE 0 OF WORD
            LMJ       X6,$+1            . BYTE 5 OF NEXT PRECEDING WORD
            ANX,U     X8,1              . WORD = WORD - 1
            LXM,U     A3,1              . A3 = BYTE 4 FOR NEXT CALL
            SA,S6     A6,0,X8           . COMPLETE THIS CALL (STORE BYTE 5, NOW)
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

SUBRTN DICK HOGERS' UTILITY SUBROUTINE PACKAGE

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...
                J        0,X6              . RETURN TO LOCATION OF EX + 1
         /.     LITERALS
          .
          .
                END
```

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

LINE # TEXT...

```
.                                                              2
. I/O TABLE 3 FOR FORTRAN-1108 SORT LINKAGES                   3
.                                                              4
$(1).                                                          5
SOPEN3*   LA        12,(DRUML,DRUM)
       ER      CSF$
          L,14      12,DRMNAM
          ER        FACIL$
       L,14    12,TABLE
          LR,14     65,3
         J          $BUILD.          GO TO BUILD COMPLETE TABLE     8
.                                                              9
. TABLE - CORE 16,000,  DRUM 200,000,  NO TAPES               10
.                                                             11
F         FORM      6,12,18.                                  12
TABLE      F         020,0,16371
         F          077,0,CORE.                               15
           F         023,0,DRMNAM+2
.                                                             18
. RESERVE CORE AREA                                           19
.                                                             20
$(0).                                                         21
CORE      RES       16371
DRUM.
          '@ASG,T  XA,F///200'
DRUML     EQU       $-DRUM
DRMNAM    'XA       '
          RES       7
          END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE TMTCSC(NWORDS,ITO,IFROM)
      DIMENSION ITO(1), IFROM(1)
      DO 1 I = 1, NWORDS
      ITO(I) = IFROM(I)
    1 CONTINUE
      RETURN
      END
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
               AXR$
               ERU$ .
       $(0)
       P         PROC
       SAVE*     NAME                                .
       K         DO P(1) . S    P(1,K),SV+K-1        .
                 GO         OUT                      .
       REST*     NAME                                .
       L         DO P(1) . L    P(1,L),SV+L-1        .
       OUT       NAME                                .
                 END                                 .
       PKT     ISOT     '28       ',R$ 252,BUF
       BUF     RES      252
       WRDCNT  RES      1
       BLOCK   RES      1
       SV      RES      5
       MESAGE           'UU0000 IS INVALID. BLOCK/WORD          DECODE'
       ERROR            '       = STATUS. SEE UP4144 PAGE C-17'
       UNEXPT           'UNEXPECTED END OF FILE ON REPORT FILE'
       SX11        +        0
       NEED      +      0
       TEST             'BEGIN REPORT'
       BLONUM      +        0
       BLANK            '       '
       WB               'REWIND'
               +        0
       PF      FORM     12,6,18
       $(1)
       THREAD*
               SAVE     A0,A2,A5,X2,X3 . SAVE CONTENTS OF CONTROL REGISTERS
               L        A5,NEED         . RETREIVE NEED NEW BLOCK FLAG
               TZ       A5              . IF NEEDED, CONTINUE
               J        DECODE          .     ELSE JUMP TO DECODE SDF CONTROL WORD
       RENTER  SZ       WRDCNT
       OK1     LMJ      X2,READ         . JUMP TO READ A NEW BLOCK
               TZ       A2              . TEST STATUS WORD
               J        ERR             .    IF NON-ZERO JUMP TO FIND CAUSE
               LX,U     X3,1            . SET X3 = 1
               SX       X3,NEED         . CANCEL NEED FOR NEW BLOCK
       DECODE  LX       X3,WRDCNT       . RETREIVE NUMBER OF WORDS PROCESSED
       .                 THIS WILL ALSO SET POINTER TO SDF CONTROL WORD
               L,S1     A2,BUF,X3       . RETREIVE SDF CODE FROM CONTROL WORD
               TNZ      A2              . IF NOT A DATA IMAGE, CONTINUE
               J        DATA            .     ELSE JUMP TO PROCESS DATA
               TNE,U    A2,050          . IF NOT A LABEL IMAGE, CONTINUE
               J        LABEL           .     ELSE JUMP TO BYPASS LABEL
               TNE,U    A2,077          . IF NOT END-OF-FILE, CONTINUE
               J        EOFXIT          .     ELSE TAKE EOF EXIT
               TNE,U    A2,040          . IF NOT PAD RECORD, CONTINUE
               J        RENTER          .     ELSE JUMP TO RE-ENTER SUBROUTINE
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
        LA      A2,BUF,X3      . RETREIVE ENTIRE CONTROL WORD FOR MESSAGE
        S       A2,MESAGE      . PREPARE ERROR MESSAGE
            SX      X11,SX11           . SAVE X11
            LMJ     X11,MOVBIN         . CALL DICK'S MOVE/CONVERT ROUTINE
            PF      1,0,MESAGE+5       . FORMS CONTROL W/ARGS
            PF      0,0,BLONUM         . FORMS CONTROL W/ARGS
            LMJ     X11,MOVBIN         . CALL DICK'S MOVE/CONVERT ROUTINE
            PF      2,0,MESAGE+6       . FORMS CONTROL W/ARGS
            PF      0,0,WRDCNT         . FORMS CONTROL W/ARGS
            LX      X11,SX11           . RESTORE X11
            LA      A0,(PF 1,8,MESAGE) . LOAD ERROR MESAGE
            ER      PRINT$             . PRINT THE MESSAGE
            LA      A2,BLONUM          . GET BLOCK NUMBER
            TNE,U   A2,0               . IF BLOCK NUMBER NOT = 0, CONTINUE
            J       RENTER             .    ELSE JUMP TO RE-ENTER
            TNE,U   A2,1               . IF BLOCK NE 1, CONTINUE
            J       OK1                .    ELSE JUMP TO OK1
            TNE,U   A2,2               . IF BLOCK NUMBER NOT = 2, CONTINUE
            J       OK1                .    ELSE JUMP TO OK1
        LMJ     X2,READ        . JUMP TO READ A NEW BLOCK
        TNE,U   A2,1           . IF NOT END OF FILE, ABORT WITH MESSAGE
        J       EOFXIT         .    ELSE TAKE END OF FILE RETURN
        ER      ERR$           . STOP RUN
LABEL   AX,S2   X3,BUF,X3      . ADD LABEL LENGTH TO WORDS PROCESSED
        AX,U    X3,1           . INCREMENT FOR LABEL CONTROL WORD
        SX      X3,WRDCNT      . STORE NEW WORD COUNT
        J       DECODE         . SKIP THIS LABEL IMAGE
ERR     TNE,U   A2,1           . IF NOT END OF FILE, CONTINUE
        J       EOFXIT         .    ELSE RETURN THRU EOF EXIT
ERR1    SX      X11,SX11       . SAVE X11
            LMJ     X11,BINOCT         . MOVE & CONVERT TO FD OCTAL
            PF      1,0,ERROR          . MOVE TO ERROR MES
            PF      0,0,A2             . PICK UP ERROR CODE
            LX      X11,SX11           . RESTORE X11
        L       A0,(PF 1,7,ERROR) . LOAD ERROR MESSAGE
        ER      PRINT$         . PRINT IT
        ER      ERR$           . STOP
EOFXIT  REST    A0,A2,A5,X2,X3         . RESTORE REGISTERS FOR RETURN
        J       4,X11                  . RETURN THRU EOF ADDRESS
DATA    L,S2    A2,BUF,X3      . RETREIVE LENGTH OF DATA IMAGE
        TNZ     A2             . IF NOT ZERO, CONTINUE
        J       LENGTH         .    ELSE JUMP TO LENGTH ERROR MESSAGE
        S       A2,*0,X11      . TRANSMIT LENGTH TO CALLING PROGRAM
        AX,U    X3,1           . ADJUST POINTER TO LOCATE FIRST IMAGE WORD
        A       A2,X3          . ADD IMAGE LENGTH TO WORD COUNT
        TG      A2,BLOCK       . IF BLOCKSIZE EXCEEDS WORD COUNT, CONTINUE
        J       SPAN           .    ELSE JUMP TO SPANNED RECORDS
        S       A2,WRDCNT      . SAVE NEW WORD COUNT
SEND    LR      R1,*0,X11      . RETREIVE LENGTH INTO REPEAT COUNTER
        LX      X2,1,X11       . STORE ADDRESS OF RECEIVING ARRAY
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: ASSEMBLER

```
LINE #   TEXT...

 101            LXI,U   X2,1              . ESTABLISH INCREMENTATION OF SUBSCRIPTS
 102            AX      X3,(1,BUF)        . ESTABLISH SENDING ADDRESSES
 103            BT      X2,,*X3           . TRANSMIT THE IMAGE
 104    CHECK   L       A2,*0,X11         . RETREIVE IMAGE LENGTH
 105            TE,U    A2,3              . IF EQUAL TO 3, CONTINUE
 106            J       NORMAL            .    ELSE TAKE NORMAL RETURN
 107            L       A2,*1,X11         . RETREIVE FIRST WORD OFIMAGE
 108            TNE     A2,TEST           . IF IT IS NOT = 'BEGIN ', CONTINUE
 109            J       TEST2             .    ELSE JUMP TO ANOTHER TEST
 110    NORMAL  REST    A0,A2,A5,X2,X3            . RESTORE REGISTERS FOR RETURN
 111            J       6,X11                     . RETURN TO CALLER
 112    TEST2   LX      X3,1,X11          . RETREIVE WORD 2 OF IMAGE
 113            L       A2,1,X3           .    AND PLACE IN A2
 114            TE      A2,TEST+1         . IF IT IS EQUAL TO 'REPORT', CONTINUE
 115            J       NORMAL            .    ELSE JUMP TO NORMAL RETURN
 116            L       A2,2,X3           . RETREIVE 3RD WORD = J SEQUENCE NUMBER
 117            S       A2,*2,X11         . STORE SEQUENCE NUMBER
 118            REST    A0,A2,A5,X2,X3            . RESTORE REGISTERS FOR RETURN
 119            J       3,X11             . TAKE NEW REPORT RETURN
 120    LENGTH  LA,S4   A2,BUF,X3 .
 121            TNE,U   A2,010 .
 122            J       BLINE
 123            TNE,U   A2,030
 124            J       EOFXIT
 125            TNE,U   A2,040
 126                                      . IF FORTRAN CODE NOT A PAD IMAGE, CONTINUE
 127            J       READNI            .  (SEE COMMENTS CIRCA LINE 150)
 128            LA      A2,BUF,X3         .    JUMP TO READNX TO HANDLE PAD IMAGE (6FIX X3)
 129            SA      A2,MESAGE         . RETREIVE SDF CONTROL WORD
 130               SX        X11,SX11     . SETUP ERROR MESSAGE
 131               LMJ       X11,MOVBIN           . SAVE X11
 132               PF        1,0,MESAGE+5         . CALL DICK'S MOVE/CONVERT ROUTINE
 133               PF        0,0,BLONUM           . FORMS CONTROL W/ARGS
 134               LMJ       X11,MOVBIN           . FORMS CONTROL W/ARGS
 135               PF        2,0,MESAGE+6         . CALL DICK'S MOVE/CONVERT ROUTINE
 136               PF        0,0,WRDCNT           . FORMS CONTROL W/ARGS
 137               LX        X11,SX11             . FORMS CONTROL W/ARGS
 138               LA        A2,('LENGTH')        . RESTORE X11
 139               SA        A2,MESAGE+7
 140            LA      A0,(PF 1,8,MESAGE)        . IDENTIFY PLACE OF ERROR
 141            ER      PRINTS            . ESTABLISH PACKET ADDRESS
 142               TG,U      A2,3         . PRINT THE MESSAGE
 143            ER      ERRS                      . IF C(A2) IS LESS OR EQUAL 2 RENTER
 144               J         RENTER       . STOP
 145    BLINE   LA      A2,BLANK          . SET A2 EQUAL TO WORD OF BLANKS
 146            SA      A2,*1,X11         . STORE BLANKS INTO LINE IMAGE
 147            LA,U    A2,1              . SET A2 = 1
 148            SA      A2,*0,X11         . SET IMAGE LENGTH = 1
 149            AX,U    X3,1              . INCREMENT POINTER TO CONTROL WORD
 150            SX      X3,A2             . SET A2 = POINTER
```

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: ASSEMBLER

LINE #   TEXT...

```
          TG      A2,BLOCK       . TEST POINT GREATER THAN BLOCKSIZE
          SZ      NEED           . IF SO, SET NEED NEW BLOCK FLAG
          SX      X3,WRDCNT .
          J       NORMAL
SPAN      TNE     A2,BLOCK       . IF WORD COUNT IS NOT = BLOCKSIZE CONTINUE
          J       READNX         .     ELSE JUMP TO SET NEED TO READ BLOCK
          ANA     A2,BLOCK       . WORD COUNT - BLOCKSIZE = SPAN LENGTH
          SA      A2,WRDCNT       . SPAN LENGTH = WORD COUNT OF NEXT BLOCK
          L       A5,*0,X11      . RETREIVE LENGTH
          ANA     A5,A2          . LENGTH - SPAN = FIRST PORTION LENGTH
          LR      R1,A5          . LOAD REPEAT COUNTER WITH THAT LENGTH
          LX      X2,1,X11       . LOAD ARRAY ADDRESS
          LXI,U   X2,1           . LOAD SUBSCRIPT INCREMENT
          AX      X3,(1,BUF)     . LOAD BUFFER ADDRESS
          BT      X2,,*X3        . TRANSMIT IMAGE
          LMJ     X2,READ        . JUMP TO READ NEXT BLOCK
          TZ      A2             . IF SUCCESSFUL READ, CONTINUE
          J       BAD            .     ELSE JUMP TO BAD BLOCK
          LR      R1,WRDCNT      . LOAD SPAN LENGTH INTO REPEAT COUNTER
          LX      X2,1,X11        . LOAD ARRAY ADDRESS
          AX      X2,A5          . BIAS ADDRESS FOR SECOND PORTION TRANSFER
          LXI,U   X2,1           . LOAD SUBSCRIPT INCREMENT
          LX      X3,(1,BUF)     . LOAD BUFFER ADDRESS
          BT      X2,,*X3        . TRANSMIT IMAGE
          J       CHECK          . JUMP TO CHECK FOR SENTINEL
BAD       TE,U    A2,1           . IF END OF FILE READ, CONTINUE
          J       ERR1           .     ELSE JUMP TO TAPE STATUS ERROR
          L       A0,(PF 1,7,UNEXPT)  . LOAD ERROR MESSAGE
          ER      PRINT$         . PRINT IT
          ER      ERR$           . STOP
READN1    AX,U    X3,1            . FUDGE X3 FOR NEXT OPERATION
READNX    SZ      NEED           . SET NEED NEW BLOCK FLAG
          LA      A2,X3           . RETREIVE POINTER
          ANA,U   A2,1            . RESTORE POINTER TO CONTROL WORD
          LA,S4   A5,BUF,A2       . ACCESS FORTRAN CONTROL BYTE
                                  .     010 = "C" = NORMAL DATA IMAGE
                                  .     030 = "S" = END-OF-FILE
                                  .     040 = "(" = PAD IMAGE TO FILL OUT TAPE BLOCK
          TNE,U   A5,040          . IF CODE NOT A PAD IMAGE, CONTINUE
          J       RENTER          .   ELSE JUMP TO RE-ENTER & GRAB A BLOCK
          J       SEND           . JUMP TO SEND IMAGE TO CALLER
READ      L,U     A0,PKT         . SETUP THE I/O PACKET
          ER      IOW$           . ISSUE THE READ AND WAIT COMMAND
              LA        A2,BLONUM          . GET BLOCK NUMBER
              AA,U      A2,1               . INCREMENT BLOCK NUMBER
              SA        A2,BLONUM          . RESTORE NEW BLOCK NUMBER
          L,H2    A2,PKT+3       . RETREIVE NUMBER OF WORDS ACTUALLY READ
          S       A2,BLOCK       . STORE THIS BLOCKSIZE
          L,S1    A2,PKT+3       . RETREIVE I/O STATUS CODE
          J       0,X2           . RETURN TO CALL LINE + 1
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

LINE #   TEXT...

```
REWIND•
        SZ       NEED                  . SET NEED NEW BLOCK FLAG
           SZ         BLONUM                     . SET ERROR BLOCK NUMBER TO ZERO
        SX,H2    X11,WB+1 .
        LMJ      X11,NREWS .
        +        0100,(28) .
        +        $-2-TREAD,WB .
        LX,H2    X11,WB+1 .
        J        1,X11                 . RETURN TO CALLING PROGRAM
          END                                  .
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

ZERO   STORE FLOATING POINT ZEROES INTO A SPECIFIED ARRAY

PROGRAM TYPE: SUBROUTINE   PROGRAM LANGUAGE: FORTRAN

LINE #   TEXT...

```
      SUBROUTINE ZERO(ARRAY,NWORDS)
      DIMENSION ARRAY(1)
      DO 1 I = 1, NWORDS
      ARRAY(I) = 0.0
    1 CONTINUE
      RETURN
      END
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

PROGRAM TYPE: SUBROUTINE PROGRAM LANGUAGE: FORTRAN

LINE # TEXT...

```
      SUBROUTINE ZEROI(IRRAY,NWORDS)
      DIMENSION IRRAY(1)
      DO 1 I = 1, NWORDS
      IRRAY(I) = 0
    1 CONTINUE
      RETURN
      END
```

```
THIS ELEMENT IS AN INDEX FILE USED TO PRINT THE PROGRAM ROSTER.
CARD COLUMN 1 IS A PROGRAM TYPE CODE:
                1 = MAIN PROGRAM
                2 = SUBPROGRAM
                3 = SUBROUTINE FROM THE LIBRARY
                4 = SYMBOLIC TEXT ELEMENT (LIKE THIS ELEMENT)
CARD COLUMN 3 IS A PROGRAM LANGUAGE CODE:
                1 = FORTRAN
                2 = ASSEMBLER
                3 = COBOL (NOT USED)
                4 = SYMBOLIC (NOTES OR DATA)
CARD COLUMNS 5 - 10 IDENTIFIES THE MAIN PROGRAM NAME OR A CODE FOR
                THE NOTES OR DATA TO FOLLOW.
CARD COLUMNS 12 - 17 IDENTIFIES THE "DECK" OR ELEMENT NAME WHICH IS
                ACTUALLY BEING PRINTED.
CARD COLUMNS 21 - 80 IS THE DESCRIPTION OR TITLE OF THE ELEMENT.


  STANDARD COMPREHENSIVE OCCUPATIONAL DATA ANALYSIS PROGRAMS (CODAP)

4 4 INTRO  INTRO    INTRODUCTION TO CODAP
4 4 PHASES PHASES   PHASES OF A CODAP STUDY
1 1 ASFACT ASFACT   ANALYSIS OF SECONDARY FACTORS
1 1 AUTOJT AUTOJT   AUTOMATIC JOB TYPE GENERATOR
1 1 AVALUE AVALUE   AVERAGE VALUE PROGRAM
1 1 AVGPCT AVGPCT   AVERAGE BY PER CENT MEMBERS PERFORMING
1 1 CORREG CORREG   CORRELATIONS AND REGRESSIONS
2 1 CORREG ALPHA    LOADER OF ALPHA CHARS SUBROUTINE (FROM PERSUB)
2 1 CORREG CORRLB   CORRELATION SUBROUTINE (FROM PERSUB)
2 1 CORREG CRTOTP   CORE TO TAPE SUBROUTINE (FROM PERSUB)
2 1 CORREG PLEVEL   PROBABILITY LEVEL SUBROUTINE (FROM PERSUB)
2 1 CORREG POSTAP   POSTION TAPE SUBROUTINE (FROM PERSUB)
2 1 CORREG PRIMSC   PRINT MEANS,SD,&CORR SUBROUTINE (FROM PERSUB)
2 1 CORREG PRINT1   PRINT SUBROUTINE (FROM PERSUB)
2 1 CORREG REGREF   REGRESSION SUBROUTINE (FROM PERSUB)
2 1 CORREG TAPGEN   TAPE GENERATION SUBROUTINE (FROM PERSUB)
2 1 CORREG TPTOCR   TAPE TO CORE SUBROUTINE (FROM PERSUB)
2 1 CORREG ZEROST   ZERO OUT ARRAY SUBROUTINE (FROM PERSUB)
1 1 DIAGRM DIAGRM   DIAGRAM OF THE CLUSTERED GROUPS
2 1 DIAGRM STRT     SELECT STARTER GROUPS FOR THE DIAGRAM
2 1 DIAGRM I2FA     CONVERT FLOATING POINT TO FIELDATA
1 1 DIST2X DIST2X   TWO WAY DISTRIBUTION PROGRAM
3 1 DIST2X COMPA    COMPARE ALPHANUMERIC VALUES
3 1 DIST2X COMPC    COMPARE COLUMN VALUES IN PROPER MODE
3 1 DIST2X COMPN    COMPARE NUMERIC VALUES
3 1 DIST2X COMPRC   COMPARE ROW/COLUMN ORDERED PAIR VALUES
3 1 DIST2X DISTX1   SEARCH FOR PROPER INTERVAL IN THE DISTRIBUTION
3 1 DIST2X FORM2X   GENERATE FORMATS FOR DIST2X PRINTING PHASE
3 1 DIST2X OVRFLO   HANDLE DIST2X CORE TABLE OVERFLOW/ DISK MERGE
3 1 DIST2X SHIFT    SHIFT COLUMN TITLES IN DIST2X FOR RIGHT JUSTIFICATION
```

COMPUTATIONAL SCIENCES DIVISION
AF HUMAN RESOURCES LABORATORY
AIR FORCE SYSTEMS COMMAND

1	1	DUVARS	DUVARS	GENERATE AND REPORT DUTY VARIABLES
1	1	EXTRCT	EXTRCT	EXTRACT REPORTS FROM THE REPORT FILES
2	1	EXTRCT	PRINTR	PRINT REPORTS EXTRACTED FROM REPORT FILE
2	1	EXTRCT	EDTPRT	EDIT AND PRINT REPORTS FROM REPORT FILE
3	1	EXTRCT	SETUPS	FILE ASSIGNMENT ROUTINE FOR EXTRCT
1	1	GRMBRS	GRMBRS	GROUP MEMBERSHIP PROGRAM
1	1	GROUP	GROUP	GROUP CLUSTERING PROGRAM
1	1	GRPDIF	GRPDIF	GROUP DIFFERENCES PROGRAM
1	1	GRPSUM	GRPSUM	GROUP SUMMARY PROGRAM
1	1	INPSTD	INPSTD	INPUT STANDARD PROGRAM
3	2	INPSTD	RUNID	RETURNS RUNID AND ACCOUNT CODE FOR CURRENT RUN
3	1	INPSTD	ZERBLK	CHANGE ZEROES TO BLANKS ON TASK/DUTY TITLE CARDS
1	1	JDINDX	JDINDX	INDEX A JOB DESCRIPTION FILE
1	1	JOBGRP	JOBGRP	JOB DESCRIPTIONS OF GROUPS AT GIVEN STAGES
1	1	JOBIND	JOBIND	JOB DESCRIPTIONS FOR INDIVIDUALS
3	1	JOBIND	REPEN1	CLOSE REPORTS & PUNCH SPECIAL JOBIND REQUEST CARD
1	1	JOBINV	JOBINV	JOB INVENTORY FOR TASKS AND DUTIES
1	1	JOBSPC	JOBSPC	JOB DESCRIPTIONS OF SPECIAL GROUPS
1	1	KPATH	KPATH	RE-ORDER HISTORY FILE INTO KPATH FILE
1	1	MTXPRT	MTXPRT	PRINT OVERLAP MATRIX OF GIVEN GROUPS
2	1	MTXPRT	OVLAP	COMPUTE OVERLAP OF GIVEN GROUPS
2	1	MTXPRT	DISKLD	LOAD DISK FOR MTXPRT PROGRAM
1	1	OVRLAP	OVRLAP	OVERLAP TASK DATA AND CREATE MATRIX
3	2	OVRLAP	RFILE	RANDOM ACCESS CASE FILE OF NON-ZERO TASK RESPONSES
1	1	PRDICT	PRDICT	PRINT THE VARIABLE DICTIONARY
1	1	PRIJOB	PRIJOB	SELECT PRIMARY JOB IDENTIFIERS
1	1	PROGEN	PROGEN	GENERATE A FORTRAN PROGRAM FOR CODAP USAGES
3	2	PROGEN	SISO	SYSTEM FILE/ELEMENT INTERFACE ROUTINE
1	1	PRTVAR	PRTVAR	PRINT INFORMATION FROM A HISTORY/KPATH FILE W/SORT
1	1	REXALL	REXALL	INTER-RATER RELIABILITY W/ALL OPTIONS
1	1	SETCHK	SETCHK	CHECK CASE DATA SETUP
1	1	TSETUP	TSETUP	SETUP JOB DESCRIPTION AND/OR REPORT FILES
1	1	TSKGRP	TSKGRP	TASK INDEX PROGRAM ON RESTRICTED GROUP OF TASKS
1	1	TSKNDX	TSKNDX	TASK INDEX PROGRAM
1	1	VARGEN	VARGEN	VARIABLE GENERATING PROGRAM
2	1	VARGEN	APLWTS	APPLY WEIGHTS TO RATINGS
2	1	VARGEN	DICTRW	DICTIONARY READ/WRITE SUBROUTINE
1	1	VARMEN	VARMEN	VARIABLE MEANS & STANDARD DEVIATIONS (CORREG W/O MATRIX)
1	1	VARSUM	VARSUM	SUMMARY OF DISTRIBUTION FOR VARIABLE VALUES
1	1	SUBLIB	SUBLIB	CODAP SUBROUTINE LIBRARY
4	4	ENTRY	ENTRY	ENTRY POINT TABLE
3	1	SUBLIB	ASGA	ASSIGN AN ALREADY CATALOGUED FILE
3	1	SUBLIB	ASGAM	ASSIGN A CYCLE OF A 3-REEL FILE
3	1	SUBLIB	ASGC	LOAD A BLANK TAPE TO BE WRITTEN ON & CATALOGUED
3	1	SUBLIB	ASGCM	LOAD BLANK TAPES TO CREATE 3-REEL "M" FILE
3	1	SUBLIB	AUTORV	AUTOMATIC FILE REVERSION: RECOVER FROM CYCLE ERROR
3	1	SUBLIB	BLANK	BLANK OUT AN ARRAY (NWORD,IARRAY)
3	2	SUBLIB	CFHIO	COBOL FILE HANDLER SUBROUTINE
3	1	FMTGEN	CPI	PACKS FORMAT CHARACTERS INTO A SINGLE WORD
3	1	SETUPS	CYCLES	HANDLE "M" FILE CYCLE ASSIGNMENTS

3	2	SUBLIB	DATACV	CONVERT FIELDATA TO BINARY, BINARY TO FIELDATA
3	1	SUBLIB	DATETM	RETURN DATE AND TIME IN NEAT FORM
3	2	SUBLIB	DELAY	CAUSE PROGRAM TO WAIT SPECIFIED # OF SECONDS
3	1	SUBLIB	DISK	SIMULATION OF IBM 7040 RANDOM ACCESS ROUTINES
3	2	SUBLIB	DKSTAT	SEE IF A MASS STORAGE FILE IS ASSIGNED
3	2	SUBLIB	DREAD	READ A FORMATTED-FORTRAN WRITTEN MASS STORAGE FILE
3	2	SUBLIB	ERII	PROVIDE OPERATOR INTERROGATION OF PROGRAM ID/STATUS
3	2	SUBLIB	FACREJ	DECODE OCTAL FACILITY REJECTION CODE TO ENGLISH
3	1	SUBLIB	FMTGEN	GENERATE FORTRAN FORMATS FROM CODAP FORMAT CARDS
3	1	SUBLIB	FORMAT	EXTRACT VARIABLE'S FORMAT FROM VARIABLE DICTIONARY
3	2	SUBLIB	FREDEV	FREE DEVICE: MAKE FORTRAN FORGET DEVICE WAS ASSIGNED
3	1	SUBLIB	FREE	FREE FILE AND ASSOCIATED DEVICE
3	1	SUBLIB	FREES	FREE FILE BUT RETAIN TAPE DRIVE FOR ANOTHER ASSIGNMENT
3	1	SUBLIB	FSORT	FORTRAN SORT SUBROUTINE / SYSTEM INTERFACE
3	1	SUBLIB	GETFIL	ISSUE ASSIGN FOR A TAPE UNTIL A DRIVE IS AVAILABLE
3	1	SUBLIB	GETMAS	ISSUE ASSIGN FOR MASS STORAGE UNTIL ACCEPTED
3	2	SUBLIB	GETPCD	RETREIVE INFORMATION FROM THE PROGRAM CONTROL TABLE (DEC)
3	1	SUBLIB	GETPCT	RETREIVE INFORMATION FROM THE PROGRAM CONTROL TABLE (OCT)
3	1	SUBLIB	H2ACND	SORT INTO ASCENDING ORDER ON H2 OF WORD
3	1	SUBLIB	H2SORT	SORT INTO DESCENDING ORDER ON H2 OF WORD
3	1	FMTGEN	INSERT	CHARACTER INSERTION SUBROUTINE
3	1	SUBLIB	IRANF	RANDOM NUMBER GENERATOR & INITIALIZATION ROUTINE
3	1	SUBLIB	JDCOPY	COPIES JOB DESCRIPTION FILE
3	1	SUBLIB	JDEOF	FINDS THE END-OF-FILE ON JOB DESCRIPTION FILE
3	1	SUBLIB	LASCMP	LOGICAL ALPHANUMERIC STRING COMPARISON
3	2	FSORT	LINK	FORTRAN SORT PACKAGE LINKAGE
3	1	SUBLIB	LISORT	POINTER LIST SORT (ASCENDING INTEGER)
3	1	SUBLIB	LSORT	POINTER LIST SORT (DESCENDING FLOATING POINT)
3	1	SUBLIB	MSKOP1	PERFORM MASK OPERATIONS ON VARIABLE MASK ARRAYS
3	1	SUBLIB	MSKOP2	PERFORM MASK OPERATIONS ON A SINGLE FIXED MASK
3	1	SUBLIB	NEXREL	FREE TAPE FILE ON A DRIVE, THEN ASSIGN NEW FILE TO IT
3	2	SUBLIB	NOHEAD	CHANGE SYSTEM DEFAULTS ON PAGE OVERFLOWS
3	2	SUBLIB	NRAND	RANDOM NUMBER GENERATOR
3	2	SUBLIB	OMSG	OPERATOR MESSAGE/INTERACTION SUBROUTINE (ON "LOAD" CONSOLE)
3	2	SUBLIB	OMSGO	OPERATOR MESSAGE/INTERACTION SUBROUTINE (ON "TD" CONSOLE)
3	1	SUBLIB	PRTDIC	PRINT THE VARIABLE DICTIONARY (PRDICT,VARGEN,PROGEN)
3	1	SUBLIB	RANDOM	STANDARD RANDOM SELECTION SUBROUTINE (IRAND,SRAND)
3	1	SUBLIB	RANF	FLOATING POINT RANDOM NUMBER FUNCTION (MUST CALL IRANF)
3	1	SUBLIB	RANSEQ	RANDOMLY SEQUENCE AN ARRAY
3	1	SUBLIB	REPEND	CLOSE REPORTS AND PUNCH A REQUEST CARD
3	1	SUBLIB	RESDRV	RESERVE A TAPE DRIVE (BEING PHASED OUT: USE FREES)
3	1	SUBLIB	ROSTER	CONTROL ROSTER & ERROR MESSAGES (PRINTC,ERRMSG,RESET)
3	1	SUBLIB	RPCOPY	COPIES A REPORT FILE FOR A NEW CYCLE
3	1	SUBLIB	RPEOF	LOCATES AN END-OF-FILE ON A REPORT FILE
3	1	SUBLIB	RPINDX	PRINTS AN INDEX OF A REPORT FILE
3	1	SUBLIB	SAMSEL	NEW SAMPLE SELECTION FOR JOBSPC,AVALUE,AVGPCT,TSKNDX
3	1	SUBLIB	SETUP	THE STANDARD FILE SETUP SUBROUTINE
3	1	SUBLIB	SETUP9	FILE ASSIGNMENT ROUTINE FOR JOBSPC & JOBGRP
3	1	SUBLIB	STATS	STATISTICS GATHERER: PRIMARILY FOR INPSTD
3	2	SUBLIB	STATUS	CHECK IF A GIVEN TAPE FILE IS ALREADY ASSIGNED

```
3 2 SUBLIB SUBHTN   DICK ROGERS' UTILITY SUBROUTINE PACKAGE
3 2 FSORT  TABL3    FORTRAN SOHT PACKAGE
3 1 SUBLIB TMTCSC   MOVE CONTENTS OF ARRAYS FROM ONE TO ANOTHER
3 2 SUBLIB TREAD    READ FORMATTED-FORTRAN WRITTEN TAPE FILES
3 1 SUBLIB ZEHO     STORE FLOATING POINT ZEROES INTO A SPECIFIED ARRAY
3 1 SUBLIB ZEROI    STORE INTEGEH ZEROES INTO A SPECIFIED ARRAY
4 4 INDEX  INDEX    INDEX FILE USED AS INPUT TO PROGHAM ROSTER ROUTINE
```